

# **ODML MONITOR**

*Version 2.0*

**User Guide**

Moscow, 2001

# 1 CONTENTS

<b>1</b>	<b><i>Contents</i></b> .....	<b>2</b>
<b>2</b>	<b><i>Introduction</i></b> .....	<b>3</b>
<b>2.1</b>	<b>Application</b> .....	<b>3</b>
<b>2.2</b>	<b>Requirements to User Skills</b> .....	<b>3</b>
<b>3</b>	<b><i>Description of Operations</i></b> .....	<b>4</b>
<b>3.1</b>	<b>Main Window</b> .....	<b>4</b>
<b>3.2</b>	<b>Main Window Menu</b> .....	<b>6</b>
<b>3.3</b>	<b><i>ODML Browser Window</i></b> .....	<b>10</b>
<b>3.4</b>	<b><i>Table Browser Window</i></b> .....	<b>12</b>
<b>4</b>	<b><i>Appendix</i></b> .....	<b>14</b>
<b>4.1</b>	<b>Values of the SQL Types</b> .....	<b>14</b>

## **2 INTRODUCTION**

### **2.1 Application**

The ODML Monitor is developed for creating, editing, and executing database queries. The queries may be written in the following languages:

- Object Query Language (OQL);
- Structured Query Language (SQL).

While using OQL, the user writes queries for an object database, the latter being in fact the object superstructure over a classic relational database. In SQL, the user writes queries for the relational database. Each OQL query is converted into an SQL query (most of the time a set of SQL queries).

### **2.2 Requirements to User Skills**

The user should have basic computer skills and study this manual. She also needs to be familiar with Pluk Language, Object To Database Mapping Language (ODML) and Structured Query Language (SQL) Reference Books. Additionally, the user should be acquainted with the Open Database Connectivity (ODBC) software that provides the standard protocol for working with a database.

## 3 DESCRIPTION OF OPERATIONS

### 3.1 Main Window

Fig. 3.1-1 shows the main window of the Monitor with an example of an OQL query.

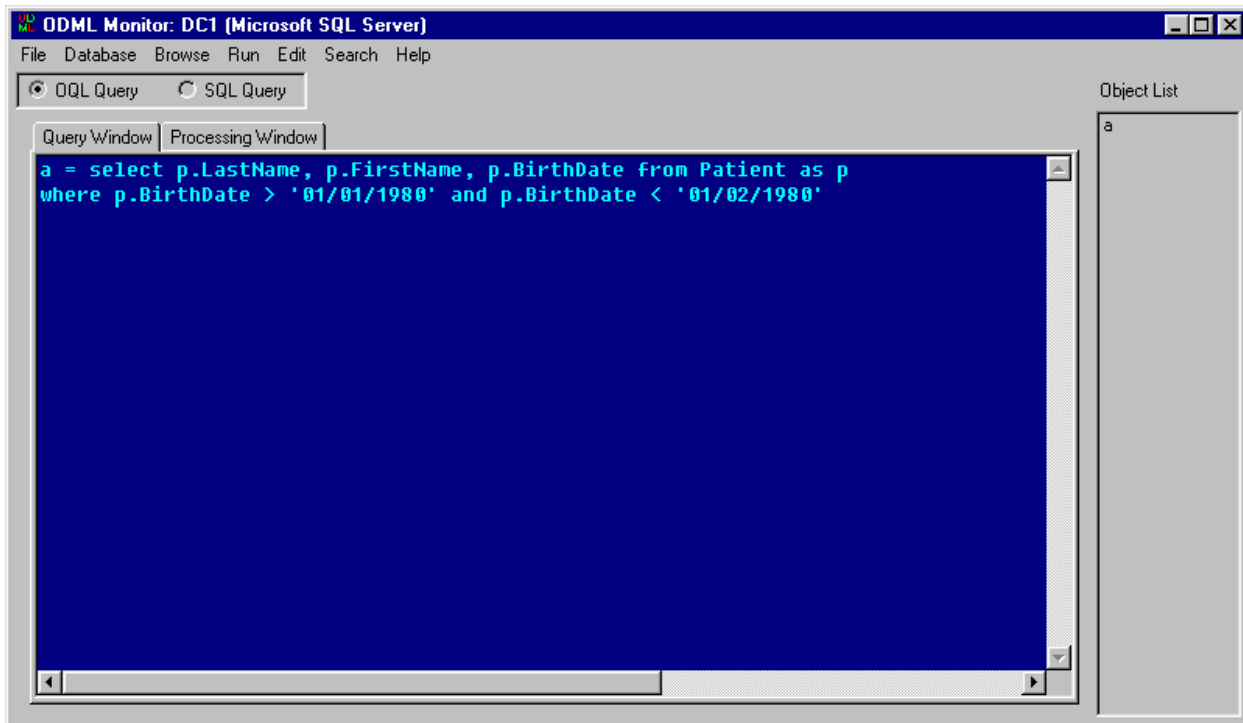


Fig. 3.1-1 Main Window

#### 3.1.1 OQL Query

If this radio button is selected, OQL queries will be executed in the *Query Window* (see 3.1.3).

#### 3.1.2 SQL Query

If the radio button is selected, SQL queries will be executed in the *Query Window* (see 3.1.3).

#### 3.1.3 Query Window

The window for entering OQL / SQL query text (Fig. 3.1-2). See the details in ODML Reference Book and SQL Reference Book.

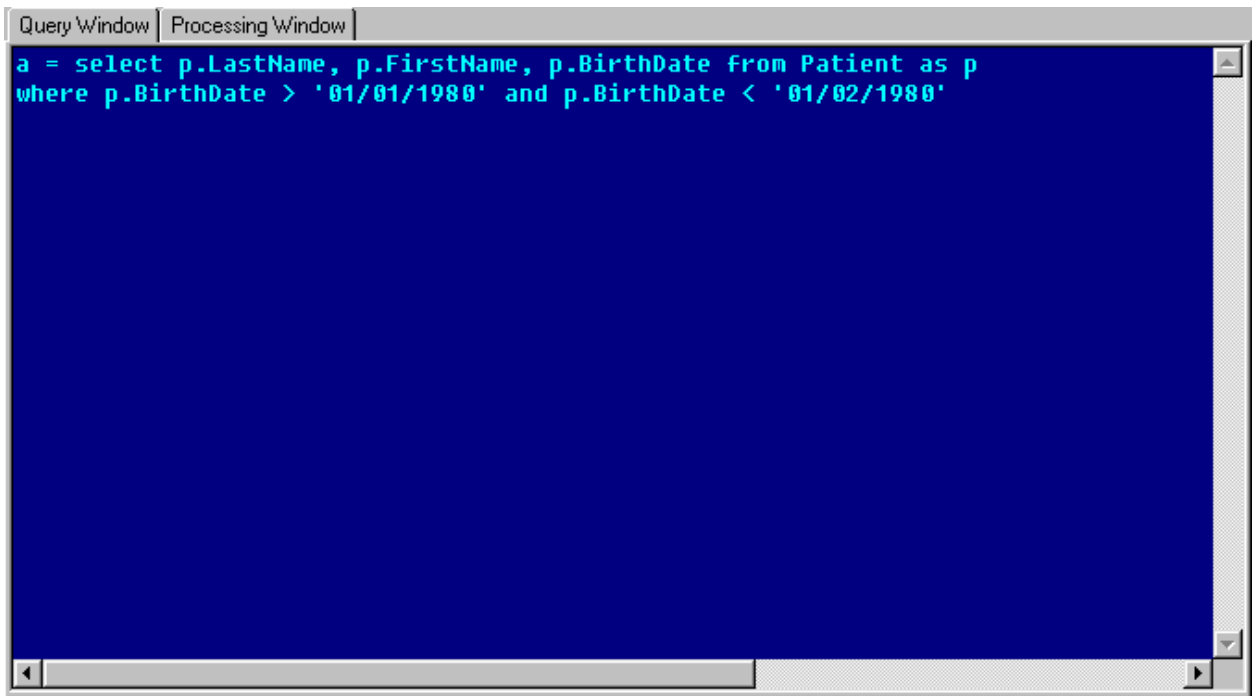


Fig. 3.1-2 *Query Window* Tab of the Main Window

### 3.1.4 *Processing Window*

The window for entering query postprocessing text in Pluk Language (Fig. 3.1-3 ). See Pluk Language Reference Book for more details.

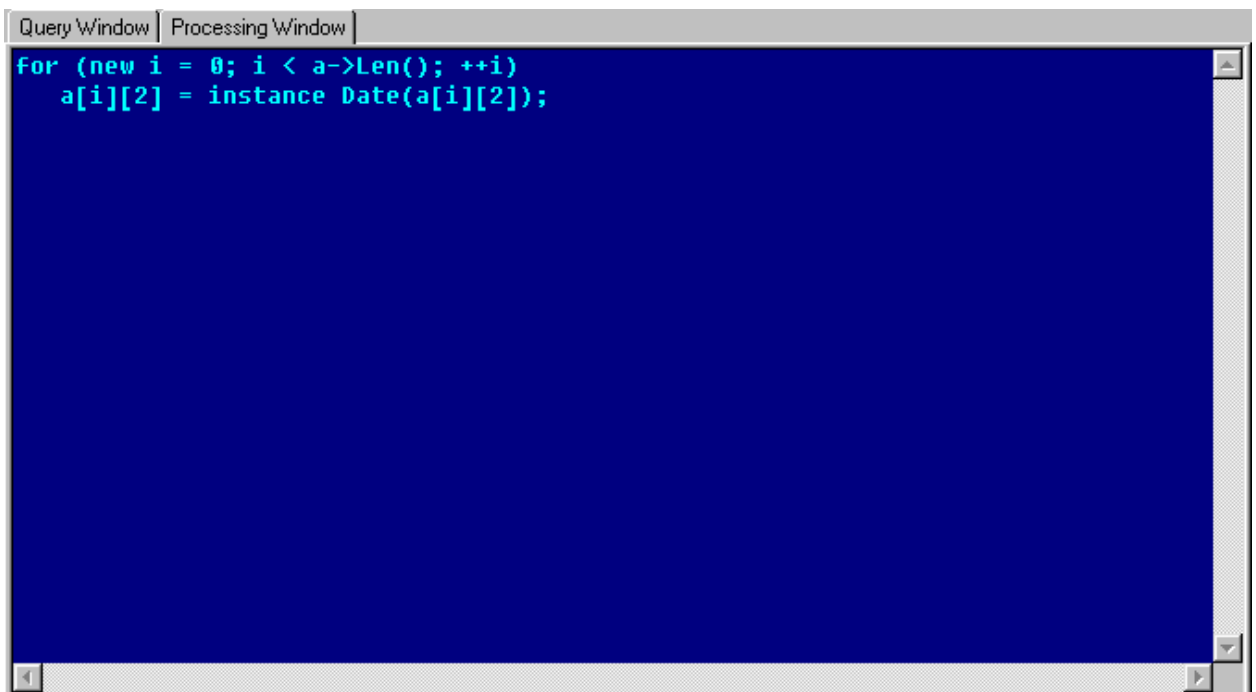


Fig. 3.1-3 *Processing Window* Tab of the Main Window

Postprocessing code can use the names of the variables created in an OQL / SQL query (see 3.1.3).

The user can create new local and global variables / functions, classes and methods of Pluk Language in a postprocessing code. In this case, the local variables / functions cease to exist upon completion of the code; the classes, methods and global variable / functions are being stored until the end of the Monitor's session.

### 3.1.5 Object List

A list of the objects created after an OQL / SQL query execution. The name of the variable at the left of the query result assignment operator has been added to the objects list (if it was not there yet). Left double-click on the name of an object opens the object browsing window (see Pluk IDE User Guide).

An OQL query execution results in an object of the type **Vector**. Each element of this object contains the inquired part of the database object that satisfies the query.

An SQL query execution results in an object of the class **SQLResult** (derived from the **Vector**). Each element of this object contains a vector of the inquired fields of the database table row that satisfies the query. The field **ColName** of an object of the class **SQLResult** is a vector that contains the names of the inquired fields. The field **ColType** is a vector that contains the types of the inquired fields as integers (see 4.1).

## 3.2 Main Window Menu

Fig. 3.2-1 shows the main window menu.



Fig. 3.2-1 Main Window Menu

### 3.2.1 File

The menu contains tools for working with database scheme files and query files (Fig. 3.2-2).

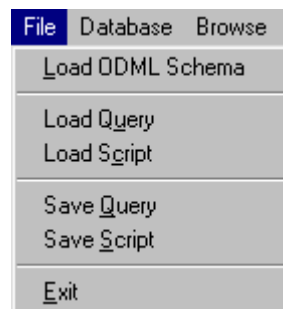


Fig. 3.2-2 File Menu

#### 3.2.1.1 Load ODML Schema

Loads a database scheme from a text file with a \*.scm extension (the scheme of mapping of the object database classes to the relational database tables) or a \*.cls extension (the scheme of the object database).

Moreover, a database scheme can be loaded from the database itself (see 3.2.2.2).

### 3.2.1.2 Load Query

Loads an OQL / SQL query text from a file with a \*.qry extension into the *Query Window* (see 3.1.3).

### 3.2.1.3 Load Script

Loads postprocessing text in Pluk Language from a file with a \*.plk extension into the *Processing Window* (see 3.1.4).

### 3.2.1.4 Save Query

Saves OQL / SQL query text in the *Query Window* (see 3.1.3) into a file with \*.qry extension.

### 3.2.1.5 Save Script

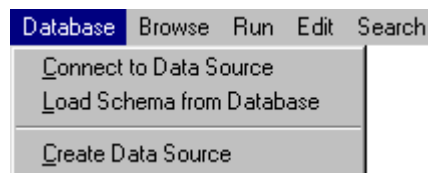
Saves postprocessing text in Pluk Language in the *Processing Window* (see 3.1.4) into a file with a \*.plk extension.

### 3.2.1.6 Exit

Terminates the Monitor's session.

## 3.2.2 Database

This menu contains tools for working with ODBC sources (Fig. 3.2-3).



**Fig. 3.2-3 Database Menu**

### 3.2.2.1 Connect to Data Source

Allows connecting to the database for which a report is needed. *Connection to Local Data Source?* pops up after selecting this item. *Yes*-click results in the Select ODBC source window and, possibly, connect window that will request the user name and password to access the database. *No*-click results in the *Host Logon* window that requires the fully qualified name or IP-address of the remote computer, user name and password for connection to the remote computer. Then the *Database Logon* window opens that requires the name of the ODBC source on the remote computer, user name and password to access the database.

If the database contains database schemes, the *ODML Schemata* window will open, upon establishing the connection, to select the scheme to be loaded.

### 3.2.2.2 Load Schema from Database

Loads a database scheme from the database itself. In this case, the *ODML Schemata* window will open to select the scheme to be loaded.

### 3.2.2.3 Create Data Source

Creates a new ODBC source. By means of the window with the ODBC drivers list (types of RDBMS), the user selects the needed RDBMS. Then in a sequence of windows specific for a

particular driver, the user sets up the source name, server name (for a client-server RDBMS) or file name (for a file RDBMS), etc.

### 3.2.3 Browse

This menu contains tools for viewing various entities in a database (Fig. 3.2-4).

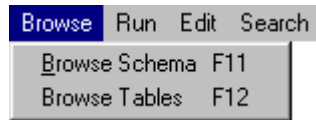


Fig. 3.2-4 Browse Menu

#### 3.2.3.1 Browse Schema

Opens the *ODML Browser* window for viewing classes and attributes of an object database (see 3.3).

Assigned key — **F11**.

#### 3.2.3.2 Browse Tables

Opens the *Table Browser* window for viewing tables and fields of a relational database (see 3.4).

Assigned key — **F12**.

### 3.2.4 Run

This menu contains tools for executing queries and postprocessing code (Fig. 3.2-5).

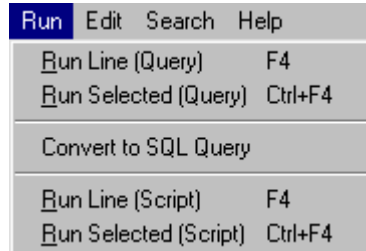


Fig. 3.2-5 Run Menu

#### 3.2.4.1 Run Line (Query)

Executes an OQL / SQL query written in one line. The line has been pointed at by the cursor in the *Query Window* (see 3.1.3).

#### 3.2.4.2 Run Selected (Query)

Executes an OQL / SQL query. The query text has been selected in the *Query Window* (see 3.1.3).

#### 3.2.4.3 Convert to SQL Query

Converts an OQL query into an SQL query. The OQL query text has been selected in the *Query Window* (see 3.1.3). The converted SQL query will be added at the end of the text.

One OQL query is often converted into a set of nested SQL queries. Please note that only the topmost SQL query (of all the nested queries) will be added at the end of the text.



#### 3.2.4.4 *Run Line (Script)*

Executes Pluk Language code written in one line. The line has been pointed at by the cursor in the *Processing Window* (see 3.1.4).

#### 3.2.4.5 *Run Selected (Script)*

Executes Pluk Language code. The code text has been selected in the *Processing Window* (see 3.1.4).

### 3.2.5 *Edit*

This menu contains tools for editing text and changing text parameters in the query and postprocessing code entry windows (see 3.1.3, 3.1.4, Fig. 3.2-6).



**Fig. 3.2-6 *Edit* Menu**

#### 3.2.5.1 *Undo*

Cancels / restores the last change in the text being edited.  
Assigned keys — ***Alt+BkSp***.

#### 3.2.5.2 *Cut*

Copies a selected text into the clipboard, deleting it from the work sheet.  
Assigned keys — ***Shift+Del***.

#### 3.2.5.3 *Copy*

Copies a selected text into the clipboard.  
Assigned keys — ***Ctrl+Ins***.

#### 3.2.5.4 *Paste*

Inserts the contents of the clipboard into the cursor position.  
Assigned keys — ***Shift+Ins***.

#### 3.2.5.5 *Font*

Opens the window for setting font parameters and color.

#### 3.2.5.6 *Color*

Opens the window for setting background color.

### 3.2.6 *Search*

This menu contains tools for finding a text fragment in the query and postprocessing code entry windows (see 3.1.3, 3.1.4, Fig. 3.2-7).



**Fig. 3.2-7 Search Menu**

### 3.2.6.1 Find

Opens the Find window for entering a fragment to be found.  
Assigned keys — **Alt+F3**.

### 3.2.6.2 Find Next

Searches for the next occurrence of the last entry in the text box.  
Assigned key — **F3**.

### 3.2.7 Help

Contains help information for the Monitor (Fig. 3.2-8).



**Fig. 3.2-8 Help Menu**

### 3.2.7.1 About

Opens the window with the version number and copyright information.

### 3.2.7.2 Help

Opens this User Guide.

## 3.3 ODML Browser Window

The user can view classes and attributes of an object database in this window. She can also insert a path in class inclusion hierarchy into query or postprocessing text (Fig. 3.3-1).

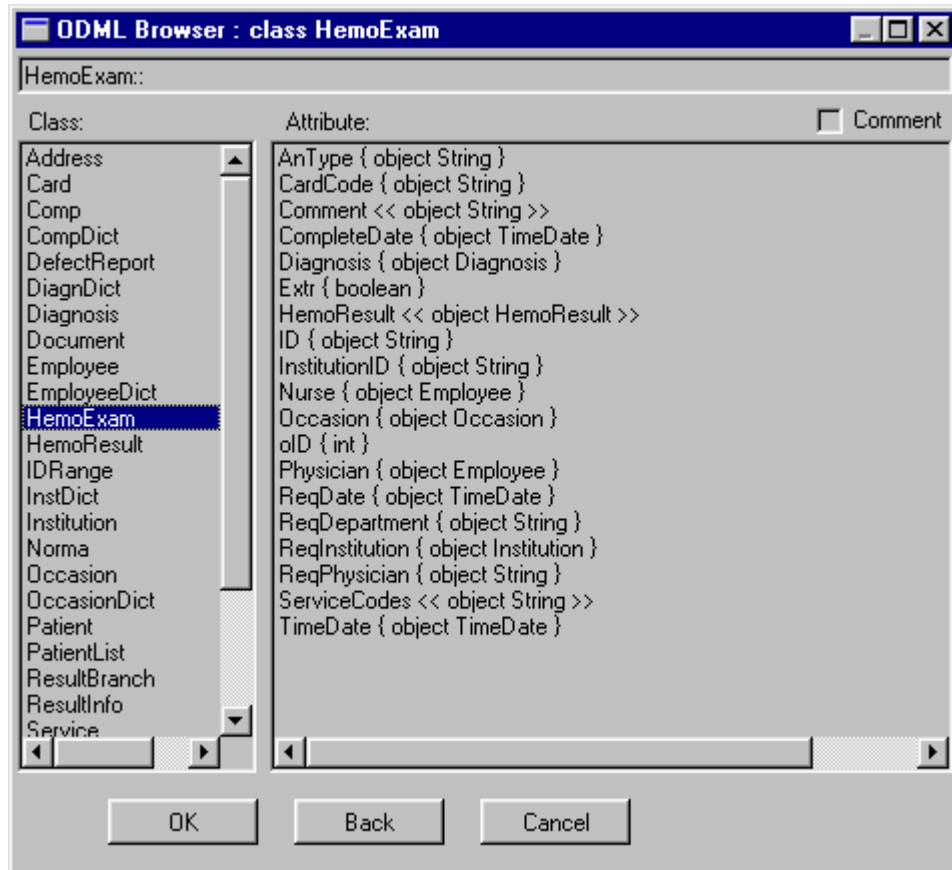


Fig. 3.3-1 ODML Browser Window

### 3.3.1 Class

A list of classes in the database.

### 3.3.2 Attribute

A list of attributes of the selected class, indicating the name and type. The type of an attribute is displayed in Pluk Language notation (see Pluk Language Reference Book). Braces indicate that the attribute stores an object of this type. Angle brackets indicate that the attribute stores a vector of an object of this type. Left double-click on an attribute makes one step down the class inclusion hierarchy.

### 3.3.3 Comment

Checking / unchecking of this box opens / closes the window with system comments on the selected class.

### 3.3.4 OK

Inserts the path selected in the class inclusion hierarchy into the cursor position in the query or postprocessing code entry window (see 3.1.3, 3.1.4).

### 3.3.5 Back

Allows one step back (to the previous window) in moving down the class inclusion hierarchy.

### 3.4 Table Browser Window

The user can view tables and fields of a relational database in this window and insert a table or field name into query or postprocessing text (Fig. 3.4-1).

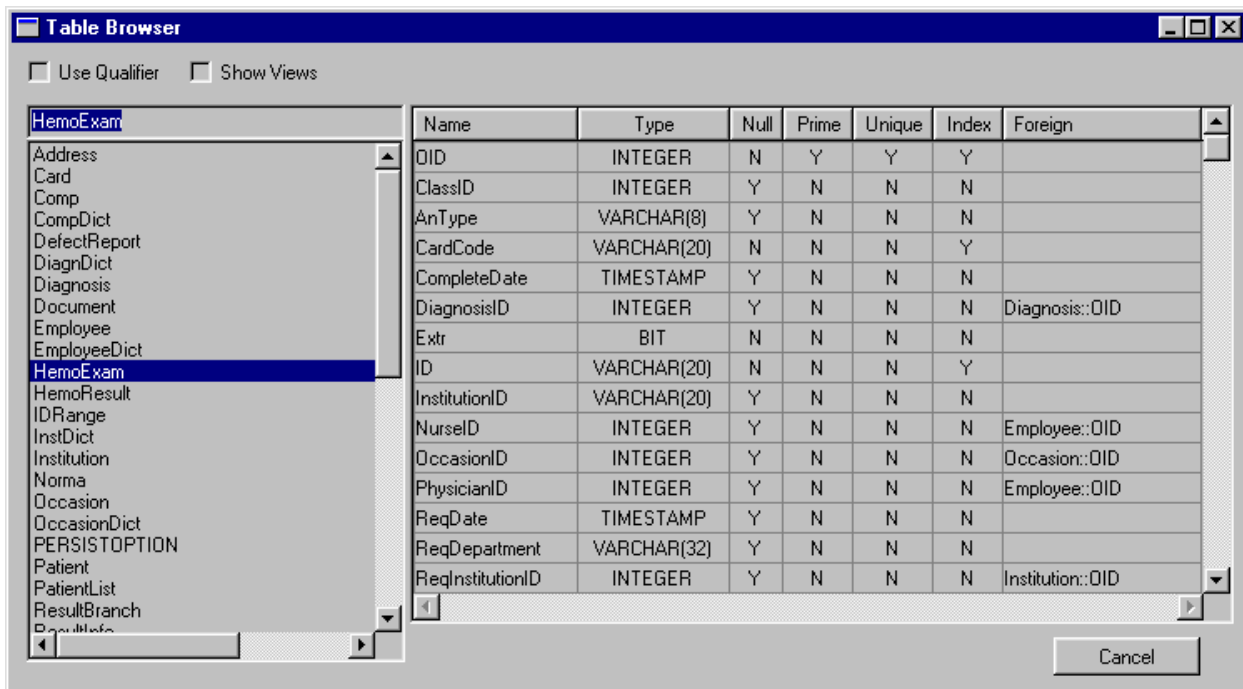


Fig. 3.4-1 Table Browser Window

#### 3.4.1 Tables List

A list of relational tables of the database.

Left double-click on a table inserts its name into the cursor position in the query or postprocessing code entry window (see 3.1.3, 3.1.4).

#### 3.4.2 Fields List

A list of fields in the table selected from the tables list. The following information is displayed for each field (see SQL Reference Book):

- the name of the field;
- the type in SQL notation;
- if the field can hold an empty value;
- if the field is a primary key;
- if the field is unique;
- if the field is indexed;
- the name of the foreign key.

Left double-click on the field inserts the names of the table and the field (separated with a period) into the cursor position in the query or postprocessing code entry window (see 3.1.3, 3.1.4).

### 3.4.3 Use Qualifier

If checked, the table name will be presented in fully qualified form: the database name, the owner's name, the table name (separated with periods, Fig. 3.4-2). Additionally, the fully qualified name will be inserted into query or postprocessing text.

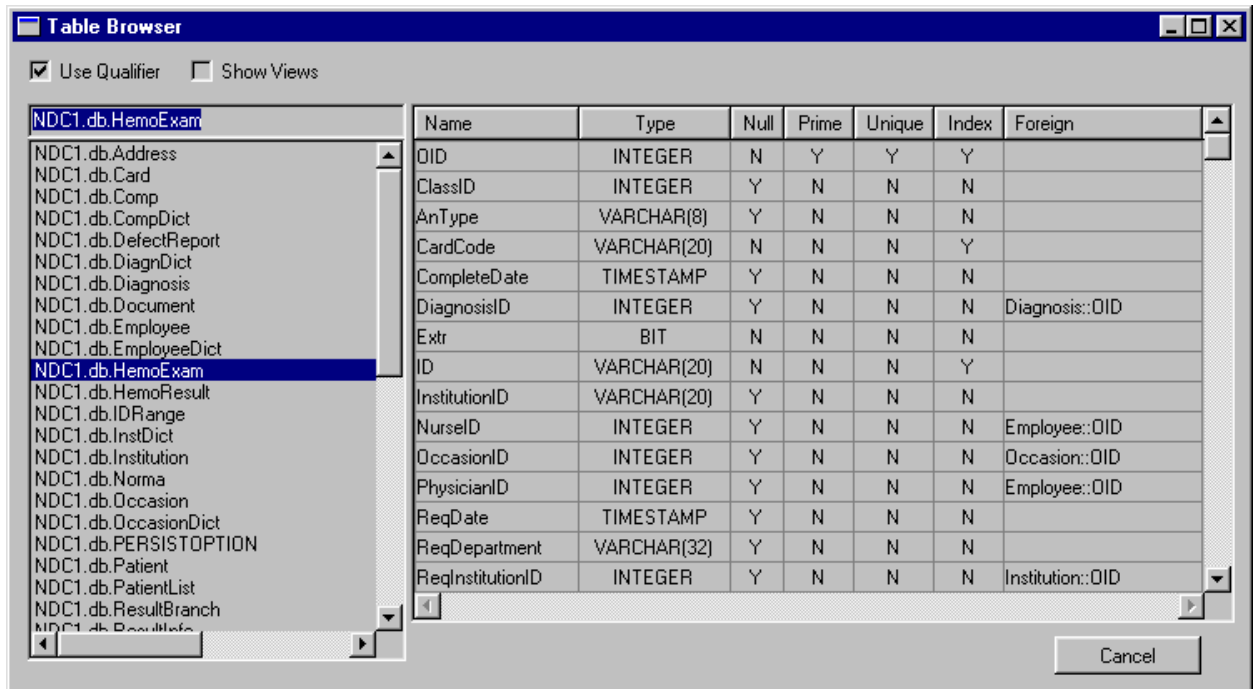


Fig. 3.4-2 Table Browser Window with Fully Qualified Table Names

### 3.4.4 Show Views

If unchecked, the tables list will contain the tables of the database. If checked, the table list will contain the views of the database, i.e. the virtual tables obtained from a query to real tables (see SQL Reference Book).

## 4 APPENDIX

### 4.1 Values of the SQL Types

Type in SQL notation	Value
NULL	0
CHAR	1
NUMERIC	2
DECIMAL	3
INTEGER	4
SMALLINT	5
FLOAT	6
REAL	7
DOUBLE	8
DATE	9
TIME	10
TIMESTAMP	11
VARCHAR	12
LONGVARCHAR	-1
BINARY	-2
VARBINARY	-3
LONGVARBINARY	-4
BIGINT	-5
TINYINT	-6
BIT	-7