

VALENTINA

written by Ruslan Zasukhin,
Paradigma (www.paradigmasoft.com)
© 1999

Getting started

Acknowledgments:

- | | |
|-----------------------|--|
| Brian Blood | - for help in development of this product. |
| Andy Bachorski | - consultant, Apple Computer, Inc. |
| Hideaki Iimori | - for help in internationalization |

Introduction

Valentina is a Relational Database Management System (RDBMS) which has an extremely fast and powerful database engine.

Valentina has its own file system, so many logical files of the database can be stored in one disk file. Valentina keeps a database in 4 separate disk files:

- “dbName.vdb” - description of the database
- “dbName.dat” - major data of the database (Tables, Relations)
- “dbName.blb” - contains BLOB-data (described below)
- “dbName.ind” - temporary data of the database (indexes, and any other temporary files)

■ **BLOB data is of variable size and can contain QuickTime movies, sound files, PICT files, etc. Basically it can hold anything that can be stored in a file on the Mac or PC.**

■ **The file “dbName.ind” can be deleted and the database will not be corrupted since Valentina will rebuild indexes when they are needed. You can use this if there is any crash during debugging.**

Database characteristics:

Max length of the disk file:	$2^{(32-1)}$ (2 GB)
Max number of Tables:	2^{32} (4'294'967'295)
Max number of Fields in the Table:	2^{16} (65'535)
Max number of records in the Table:	2^{32} (4'294'967'295)
Max size of a BLOB record:	$2^{(32-1)}$ (2 GB)

INSTALLATION

To install Valentina, just copy it to your hard disk.

If your copy is not registered, then you will get a modal dialog reminding you about registration on each open of a database. When you obtain your serial number, enter it and the dialog will not come back.

Minimal requirements:

68K version: 68020 processor, 2.5 MB of RAM, MacOS 7.5 or higher

PowerPC version: PowerPC processor, 2.5 MB of RAM, MacOS 7.5 or higher.

Enabled AppleScript.

Hard disk requirements: 1.2 MB plus the size of your databases.

The standard database operations are the following:

I Creation of database structure:

- 1) creation of a new empty database;
- 2) adding a new empty table to a database;
- 3) adding of fields (columns) to a table;

II Adding/changing contents of the tables:

- 1) adding of new records to a table;
- 2) updating existing records;
- 3) deletion of records;

III Searching/sorting:

- 1) selecting of records which match to some conditions;
- 2) sorting of selected records on one or several fields;
- 3) Looping through selected records;

IV Modification of database structure:

- 1) adding/removing of fields from tables;
- 2) changing attributes of fields (type, indexing, ...);
- 3) adding/removing of tables;

Valentina allows you to perform any of these operations using AppleScript. Let's in several short examples, see how you can do this.

The following short example shows how you can create a new database and a table with several fields:

```
set DB to make new database with data file "Customers db"
set Customer to make new base object with properties {name: "Customer"} at end of DB

tell Customer
  make new field with properties
    {name: "FirstName", type: tString, length: 20, language: "German" } at end
  make new field with properties
    {name: "LastName", type: tString, length: 20, language: "German" } at end
  make new field with properties
    {name: "BornDate", type: tDate } at end
  make new field with properties
    {name: "Phone", type: tString, length: 10 } at end
end tell
```

The next example shows how to add new records to the table:

```
tell Customer
  blank -- clears the memory buffer of the table Customer
  set fields to { "John", "Pierson", "03/05/65", "2456897" }
  make new record at end
end tell
```

In the same manner, you can update a record:

```
tell Customer
  set current record to record 1 of SomeSelection
  set fields to { "John", "Pierson", "03/05/65", "2456897" }
  -- set new values of the fields in the memory buf
  update -- updates the current record with a new values.
end tell
```

To delete a record of the table, you can write the following:

```
tell Customer
  set current record to record 1 of SomeSelection
  delete current record
end tell
```

or

```
tell Customer
  delete record 1 of SomeSelection
end tell
```

If you want to select all records of a Table then use code like:

```
set SomeSelection to select records of Customer
```

This can also be written:

```
tell Customer
  set SomeSelection to select records
end tell
```

To select records which match to some conditions you can write:

```
tell Customer
  set Selection2 to select records where
    {field "FirstName", "John",
     field "BornDate", ">=1/1/1960 and <12/31/1969"}
end tell
```

When you have a selection of records, usually you will wish to sort it on some field(s).

You can do this in the following way:

```
set Selection3 to sort Selection2 by {field "FirstName", field "LastName" }
or
set Selection3 to sort Selection2 by {field "BornDate" }
```

When you have a selection (sorted or not sorted) usually you need loop it to get values of the fields to display them or to do some calculations.

To loop through a selection you can use code like:

```
set RecsCount to get count of records in Selection3
repeat with i from 1 to RecsCount
  set current record of Customer to record i of Selection3
  set FieldsList to fields of the Customer
  -> { "John", "Pierson", "03/05/65", "2456897" }
  -- get list of values of all fields of the Customer
  -- do some work]
end repeat
```