

# **SOCRATES : tools for database design and management**

Autor(en): **Breure, Leen**

Objekttyp: **Article**

Zeitschrift: **Geschichte und Informatik = Histoire et informatique**

Band (Jahr): **2 (1991)**

PDF erstellt am: **23.05.2024**

Persistenter Link: <https://doi.org/10.5169/seals-4258>

## **Nutzungsbedingungen**

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

## **Haftungsausschluss**

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

## **SOCRATES: TOOLS FOR DATABASE DESIGN AND MANAGEMENT**

*SOCRATES: its philosophy by Leen Breure, Utrecht*

To create a database, one needs not only software in the form of a database management system but also some knowledge about its theoretical principles. One has to translate the structure of that part of the real world which contains the data, into the regularities of a database model. Especially when that reality is only perceivable through documents (e.g. historical sources), this translation process may be subjected to different interpretations and result in alternative solutions. Therefore, one may want to compare and to discuss realized database solutions. On the one hand SOCRATES offers a collection of such examples in a much more practical manner than any publication can do. On the other hand it is a software environment to create one's own database models, to adjust and document them.

SOCRATES is a set of tools (written in dBASE IV), grouped round a repository, the central storage place for the designed models. The name SOCRATES is an acronym standing for Source Oriented Creation of Relational Applications for in Time Evolving Systems, showing its connection with the relational database model and testifying that one must be able to change the database structure easily in the course of time, when new information requirements arise.

Its back-end tools make well considered database design quicker and easier. In principle they are not restricted to any specific DBMS. SOCRATES is not a graphical CASE-tool, but provides bridges to such commercial available products as SDW and EasyCASE. It is an open platform, intended to connect different kinds of software on the basis of data models. The core of SOCRATES consists of the afore-mentioned repository (= data dictionary), which is a database it self, used for storing the models of designed user databases. It supports the relational model as defined by Codd. The repository may contain several alternative designs for a certain data collection. An access program, the repository manager, provides full control of the repository's contents, supporting referential integrity, cascade deletion, automatic primary key handling et cetera.

The design process can start from sample database designs, which are

included in SOCRATES. They are easily inspected on screen or by means of printed output. After having selected the model which is most appropriate to one's needs, one may customize it in every respect. Or, if nothing comes close enough, one can start from scratch, defining database files as one used to do, and add them afterwards to the system's repository only by pushing a button, being provided with all necessary documentation for further maintenance of the database. Database models, made by means of certain graphical CASE-tools, can be imported as far as software-bridges are available.

After having completed the design process, one may push another button and generate the physical database structure corresponding to the designed model (including screen formats, indexes etc.) for any of the supported database management systems. If dBASE IV is the target system, the build-in generator may also create some Application Definition Files used by SOCRATES' front-end: a rather unique, general data input program, dBROWSER, which combines structured data entry with a documentary view on what has been stored already, providing more context than usual DBMS applications do.

This completes the development cycle: on this point there is a well documented data model and an user friendly application program for adding and modifying data. Both, model and physical database, can update each other: changes may be made to the physical database structure, which can easily be reflected by the model in the repository. Or, to the contrary, if the latter is changed, the generator can quickly modify the database structure without any loss of data.

### **SOCRATES and the research into hybrid information systems**

SOCRATES has been developed at the university of Utrecht, within the Arts Faculty as response to an increasing need of automated processing of data, embedded in especially historical documents. Although text retrieval systems may seem obvious in case of document processing, the structured nature of these factual data indicates the use of a DBMS. The documentary material suggests a data-oriented approach of the database design process and raises intriguing questions about possible generalizations of solutions found in the field of historical data processing, approaching from this angle the study of hybrid information systems (i.e. systems which combine text retrieval with structured database techniques). On this level the SOCRATES system may be regarded as a research tool to explore the following problems:

## Theory

What theoretic model adequately describes the character and development of this type of information system which contain structured data, originally embedded in documents, and function within end-user computing environments with steadily changing information requirements?

To solve this master problem, two aspects must be considered: the back-end of the system, which covers the database design, and the front-end, which regards the user application programs, used for data-entry and updating. To mention a few of the more specific problems:

### Back-end

To what degree can structured data models be used for document embedded data? Can apparently irregular document structures be reduced to regular models?

To what extent is the relational database model usable for implementation of these document models?

Is it adequate to create adaptable standard models for certain types of documents?

How usable are more integrated and more encompassing data models (with a higher degree of generalization) for a collection of different types of related documents?

Which tools will enable comprehensive assessment of such data models?

### Front-end

How can structured data entry be combined with a textual view, which provides data with context, similar to that in the original document?

What kind of documentation is needed in order to guarantee the full reusability of a filled database (also with regard to data archives)?