briefing paper

digital preservation europe

# Database Preservation: The international Challenge and the Swiss Solution

**Most administrative records are stored in databases. Today's challenge is preserving the information and making it accessible for years to come, ensuring knowledge-transfer as well as administrative sustainability. Lack of standardization has hitherto rendered the task of archiving database content highly complex. The Swiss Federal Archives have developed a new XML based format which permits long-term preservation of the relational databases content. The Software-Independent Archiving of Relational Databases (short: SIARD) offers a unique solution for preserving data content, metadata as well as the relations in an ISO conform format.**

## Why archive databases in the first place?

Long-term data preservation has always been essential for the administration. Traditionally it has assured planning and stability. Nowadays the common assumption is that computerized data is already secured. Archiving is often considered redundant as our "double-click & access" habits take over the way we think about preservation of digital records.

Yet database preservation does make sense. Firstly, in an ever-changing IT environment only archiving can truly guarantee access to data and prevent its loss. Secondly almost 85% of stored data is inactive, rendering today's databases too complex and too expensive to maintain.[1] Lastly archiving is often prescribed by law, for warranting the freedom of information (e.g. the Swiss "Öffentlichkeitsgesetz") or documenting government activities (e.g. the French "code du patrimoine"). Archiving provides a good answer to our needs, fulfilling legal requirements, facilitating data management and lowering operational costs. It is, however, a hard trade and it comes with a number of pitfalls.

## An archiving predicament, or: what should we archive?

A brief history of databases will help us gain some insights into the chief predicament of database archiving. The first databases (1960s) were organized in a clear hierarchy (1:1 or 1: n relations). This tree-like structure was prone to redundancies, necessary to allow complex relations (n: m). A century later the hierarchical model was succeeded by the network model allowing multiple relations without repetitions. Somewhat later another model was introduced, the object-oriented consisting of information clusters which represent the data. Though data could be swiftly accessed the number of queries in this model was restricted. Beyond all particular differences the above data models have one thing in common, a dependency of data and code (the software language of a given database). This strict reliance complicates data extraction from the database matrix. If we are not familiar with the software code we can hardly archive at all.

There is however an exception to the rule: the relational database. This model, introduced around 1970 by Edgar Codd, resolves the dependency of data and code. It stores all data in tables. This collection of interconnected tables permits multiple relation (n: m) and an indefinite number of queries. The use of primary keys (unique identifiers for each row) and foreign keys (references to other tables) relieves us from the need for repetitions. And even if software changes data content remains unaffected. All we have to do (in order to archive) is to extract and store the tables. Archiving relational databases is simpler, in terms of efforts and costs. Since more than 90% of all databases are relational, concentrating our efforts on their preservation is probably the best strategy. The relational model solves our main archiving predicament. This however is only the first step. The second, and perhaps the trickiest one, is finding a suitable format to ensure future access to the stored data. This is precisely what the Swiss Federal Archives (SFA) has been trying to do.

## The Swiss solution: the SIARD format

The Swiss Federal Archives were confronted with the question of database preservation since the late 1990s. Strategically the SFA decided to archive only relational databases. As part of the ARELDA project a new archiving format for relational databases was conceptualized and developed.[2] The Software-Independent Archiving of Relational Databases (short: SIARD) was introduced in 2004. Since then it has been elaborated and considerably enhanced within the PLANETS project.[3] In late summer 2008 the SFA presented a full-fledged version of the SIARD format with associated software.[4]

What does long-term preservation with SIARD mean in practical terms? The SIARD software (SIARD Suite) converts proprietary databases (MS Access, MS SQL, Oracle, etc.) into a file archive in a non-proprietary SIARD format. The SIARD archive (with the file extension .siard) represents the database in its logical form, retaining not only the primary and the metadata, but most important all the relations.

A SIARD archive is a structured non-compressed ZIP container (ZIP-64 standard), permitting practically any file size. It contains two folders: "header" and "content". The header folder stores the database context, the metadata. A single file, *metadata.xml*, assures that we can understand the technical as well as the contextual background of the database. In technical terms SIARD registers on the upmost level (the database) the identifier, the format version, the message digest code of the archiving pc terminal (verifying primary data integrity) etc. On the schema level SIARD stores lists of tables, views and routines. On the table level, SIARD records the constraints and triggers. And as we go deeper into the column level SIARD also specifies the SQL type in use, LOBs (Large Objects) names, and most important of all: foreign keys and candidate keys with referential data – i.e. the relations. At the same time SIARD contextualizes the data. On the database level it lets us register or add (with the SIARD Suite) information on the archive provenance, description, user etc. In lower levels it lets us keep details of the tables and columns names and content. This descriptive information renders the database comprehensible for future users in both contextual and technical terms.

The second folder, content, stores the primary data. The data is archived according to the database structure. For each schema SIARD automatically generates a folder (schema1, schema2, etc.), containing the corresponding table series as subfolders (table1, table2, etc.). Data itself is stored in XML files (e.g. table1.xml). This schema definition reflects the table's SQL schema metadata. And it specifies that the table is stored as a chain of lines encompassing a sequence of column entries with different XML types. BLOBs and CLOBs (Binary or Character Large Objects containing all sorts of information) are also archived. They are stored in automatically generated folders (e.g. lob1, lob2, etc.) either as TXT or BIN files (record1.text, or record1.bin, etc).

SIARD is a true mirror image of the archived database. When employing SIARD we archive both primary and metadata in a manner and form that make the database understandable and accessible. But for how long?

## SIARD and the question of long-term preservation

"Eternity is very long", said Woody Allen, "especially towards the end". In the IT world an eternity could turn out to be very short. An ephemeral life span of a format threatens long-term data accessibility. What could minimize this risk? In one word: standardization.

The use of widely accepted ISO standards ensures to a large extent that stored data could be accessed in the future. Based upon this assumption SIARD records both primary data and metadata automatically in ISO norm formats: SQL1999, UNICODE and most important of them all: XML 1.0. To ensure standardization SIARD converts all proprietary database charters into the equivalent UNICODE character set. Furthermore, SIARD does not archive synonyms as they are not a part of the standardized SQL:1999. Sticking to the standards is an iron rule.

## Last but not least

SIARD is conceived as a free file format. Its description is available on the website of the Swiss Federal Archives.[5] It does not pretend to be a Swiss army knife for archiving all database models. It is however a viable and practical solution for the long-term preservation of relational databases.

### Notes

[1] Yuhanna, Noel: "Database Archiving Remains an Important Part of Enterprise DBMS Strategy", Information & Knowledge Management Professionals (2007): ftp://ftp.software.ibm.com/software/data/sw-library/data-management/optim/reports/forrester-archiving.pdf.

[2] ARELDA is the acronym for: ARchiving of ELectronic Data

[3] http://www.planets-project.eu/

[4] The SFA employs at present a Java based application, SIARD Suite, which permits to navigate through the SIARD archive and add or update the metadata.

[5] http://www.bar.admin.ch

### References

1. Acquisition and disposition strategy of The National Archives (2007): http://www.nationalarchives.gov.uk/documents/acquisition_strategy.pdf

2. Codd, E. F., "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, vol. 13, no. 6 (1970), 377-387.

3. Code du patrimoine, July 30 2008: http://www.legifrance.gouv.fr/affichCode.do;jsessionid=2FAA76FF7AE923389AC2146821608165.tpdjo01v_2?cidTexte=LEGITEXT000006074236&dateTexte=20081001

4. Knowles, J. S. / Bell, D. M. R., "The Codasyl Model", in: Databases - Role and Structure, P. M. Stocker, P. M. D. Gray, and M. P. Atkinson (eds.) CUP, 1984.
Swiss Federal Law on Archiving (BGA), June 26 1998: http://www.admin.ch/ch/d/sr/c152_1.html

5. Swiss Federal Law on the freedom of information in the federal administration (Öffentlichkeitsgesetz, BGÖ), December 17. 2004: http://www.admin.ch/ch/d/sr/c152_3.html

Author: Amir Bernstein, *Swiss Federal Archives,*
Amir.Bernstein@bar.admin.ch