briefing paper

digital preservation europe

La conservation des bases de données: le défi international et la solution Suisse

La plupart des informations administratives sont contenues dans des bases de données. Le défi actuel est de préserver l'information, et de la rendre accessible pour les années à venir, tout en s'assurant du transfert de connaissance et de la durabilité administrative. Le manque de standardisation a jusqu'à présent rendu la tâche de l'archivage pérenne des bases de données extrêmement complexe. Les Archives Fédérales Suisses ont développé un nouveau format basé sur le XML qui permet la conservation à long terme de bases de données relationnelles. Le « Software-Independant Archiving of Relational Databases » (en abrégé : SIARD) offre une solution unique pour l'archivage des données, métadonnées ainsi que des relations, dans un format compatible avec les normes ISO.

Tout d'abord, pourquoi préserver des bases de données ?

La conservation à long terme a toujours été essentielle pour les administrations. Traditionnellement, cela leur permet planification et stabilité. Aujourd'hui, l'opinion généralement répandue veut que les données électroniques soient sécurisées. L'archivage pérenne est souvent considéré comme inutile, tant nos habitudes de « double-click et accès aux données » prennent le pas sur notre façon d'appréhender la préservation de l'information numérique.

Pourtant, la préservation des bases de données est tout à fait pertinente. Tout d'abord, dans un environnement informatique sans cesse en mouvement, seul l'archivage pérenne peut véritablement garantir l'accès à l'information et prévenir de sa perte. Ensuite, près de 85% de l'information stockée est inactive, rendant la maintenance des bases de données actuelles trop complexe et chère. [1] Enfin, l'archivage est une obligation légale, pour garantir la liberté de l'information (par exemple, la « Öffentlichkeitsgesetz » Suisse), ou documenter les activités gouvernementales (par exemple, le « Code du patrimoine » Français). L'archivage apporte une bonne réponse à ces besoins, dans la mesure où il satisfait aux nécessités légales, facilite la gestion de l'information, et diminue les coûts opérationnels. Cependant, c'est un processus difficile à mettre en place, avec son lot de pièges.

Le casse-tête de l'archivage, ou que devrions-nous archiver ?

Un bref historique des bases de données apporte quelques éclaircissements sur les principaux défis de l'archivage des bases de données. Les premières bases de données (années 1960) étaient organisées dans une hiérarchie claire (relations 1:1 ou 1:n). Cette approche arborescente favorisait les redondances, nécessaires pour mettre en place des relations complexes (n:m). Une décennie plus tard, le modèle hiérarchique était remplacé par le modèle réseau, autorisant des relations multiples sans répétition. Quelques temps plus tard, un autre modèle était présenté, celui orienté-objet, qui consiste en des clusters d'information qui représentent les données. Bien que les données puissent être facilement accessibles, le nombre de requêtes de ce modèle était limité. Au-delà de leurs différences, ces modèles ont un point commun, à savoir une dépendance entre les données et le code (langage d'une base de données). Ce lien strict complique l'extraction de l'information de la base de données, si bien que son archivage est difficile à réaliser pour celui qui n'est pas familier avec le code du logiciel.

Il y a toutefois une exception à cette règle : les bases de données relationnelles. Ce modèle, introduit autour de 1970 par Edgar Codd, résout la dépendance entre les données et le code. Il stocke toutes les données dans des tables. Une telle collection de tables interconnectées permet de relations multiples (n:m), et un nombre infini de requêtes. L'utilisation de clés primaires (identifiant unique de chaque enregistrement) et de clé étrangères (références vers d'autres tables) évite le recours aux répétitions. Et même si le logiciel change, les données ne sont pas affectées. La seule chose à faire pour archiver une base de données est d'extraire et de stocker les tables. L'archivage de bases de données relationnelles est plus simple, en termes d'effort et de coûts. Etant donné que plus de 90% des bases de données sont relationnelles, la meilleure stratégie est donc probablement de concentrer les efforts sur la préservation de ce type. Le modèle relationnel résout donc notre principal problème en ce qui concerne l'archivage, c'est pourtant seulement un premier pas. Le suivant, et peut-être le plus compliqué, est de trouver un format convenable pour s'assurer de l'accès futur aux données archivées. C'est précisément ce que les Archives Fédérales Suisses ont tenté de faire.

La solution Suisse : le format SIARD

Les Archives Fédérales Suisses ont été confrontées à la question de la préservation de bases de données dès la fin des années 1990. D'un point de vue stratégique, les AFS décidèrent de n'archiver que les bases de données relationnelles. Dans le cadre du projet ARELDA, un nouveau format d'archivage pour les bases de données relationnelles a été conçu et développé.

[2] Le « Software-Independant Archiving of Relational Databases » (en abrégé : SIARD) a été introduit en 2004. Depuis, il a été élaboré et considérablement amélioré au sein du projet PLANETS. [3] A la fin de l'été 2008, les AFS ont présenté une version complète du SIARD, ainsi que le logiciel associé. [4]

Mais que signifie, en termes pratiques, la préservation à long terme avec SIARD ? Le logiciel SIARD (SIARD Suite) permet la conversion de bases de données propriétaires (MS Access, MS SQL, Oracle, etc.) en un fichier d'archives au format non-propriétaire SIARD. L'archive SIARD (avec une extension de nom de fichier .siard) représente la base de données dans sa forme logique, incluant non seulement les données et les métadonnées mais surtout les relations.

Une archive SIARD est un conteneur ZIP structuré et non-compressé (standard ZIP-64), qui permet pratiquement toutes les tailles de fichier. Il contient deux dossiers : « header » et « content ». Le dossier d'en-tête (« header ») contient le contexte de la base de données, les métadonnées. Un seul fichier, metadata.xml, assure la compréhension des aspects techniques et contextuels de la base de données. En termes techniques, SIARD enregistre les informations du plus haut niveau de la base telles que l'identifiant, la version du format, le compte-rendu d'extraction du système effectuant l'archive et vérifiant l'intégrité des données primaires, etc. Au niveau du schéma de la base, SIARD enregistre les listes de tables, vues et procédures. Au niveau des tables, SIARD enregistre les contraintes d'intégrité et les déclencheurs (« triggers »). Et plus en détail au niveau des colonnes, SIARD spécifie également le type SQL utilisé, les noms d'objets (LOBs, « Large Objects »), et surtout les clés étrangères, les clés candidates et les données de référence associées – c'est-à-dire les relations. En même temps, SIARD fournit un contexte aux données : au niveau de la base de données, SIARD permet d'enregistrer ou d'ajouter (avec SIARD Suite) des informations sur la provenance, la description, les utilisateurs etc. de l'archive. SIARD permet également de conserver des détails de plus bas niveau, tels les noms de tables ou de colonnes, ainsi que leur contenu. Ces informations descriptives rendront la base de données compréhensible aux utilisateurs futurs, tant d'un point de vue contextuel que technique.

Le second dossier, « content », contient les données brutes. L'information est archivée selon la structure de la base de données. Pour chaque schéma, SIARD génère automatiquement un répertoire (schema1, schema2, etc.) qui contient les séries de tables correspondantes sous forme de sous-répertoire (table1, table2, etc.). Les données elles-mêmes sont stockées dans des fichiers XML (exemple : table1.xml), dont le schéma reflète la table telle qu'elle est définie dans les métadonnées du schéma SQL, et spécifie que la table est enregistrée comme une suite de lignes incluant une suite de colonnes avec des types XML différents. Les BLOBs et CLOBs (Binary ou Character Large Objects) sont aussi archivés, stockés dans des répertoires générés automatiquement (lob1, lob2, etc.) soit dans des fichiers binaires, soit dans des fichiers textes (record1.bin, record1.text, etc.).

SIARD est une véritable image miroir de la base de données archivée. Lorsque SIARD est utilisé, à la fois les données brutes – primaires – et les métadonnées sont archivées de telle manière à rendre la base de données compréhensible et accessible. Mais pour combien de temps ?

SIARD et le problème de la conservation à long terme

"Eterni« L'éternité, c'est très long, surtout à la fin », a dit Woody Allen. Dans le monde de la technologie de l'information, l'éternité pourrait s'avérer en fait très courte. La durée de vie éphémère d'un format menace l'accessibilité à long terme des données. Alors, comment minimiser ce risque ? En un mot : standardisation.

L'utilisation de standards ISO largement reconnus assure dans une large mesure que l'information stockée pourra être accédée dans le futur. Partant de cette hypothèse, SIARD enregistre à la fois les données brutes et les métadonnées dans des formats normalisés par l'ISO : SQL1999, UNICODE, et le plus important de tous – XLM 1.0. Pour s'assurer de cette standardisation, SIARD convertit tous les jeux de caractères propriétaires des bases de données dans leur équivalent UNICODE. Par ailleurs, SIARD n'archive pas les synonymes, car ils ne font pas partie du standard SQL1999. Respecter les standards est ici véritablement la règle d'or.

Pour (ne pas) conclure

SIARD est conçu comme un format de fichier libre. Sa description est disponible sur le site web des Archives Fédérales de Suisse.[5] SIARD ne prétend pas être le couteau Suisse pour l'archivage de tous les modèles de bases de données. C'est néanmoins une solution viable et pratique pour la préservation à long terme de bases de données relationnelles.

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/datamanagement/optim/reports/forresterarchiving.pdf

[2] ARELDA est l'abbréviation de ARchiving of ELectronic Data

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

[4] Les Archives Fédérales Suisses utilisent actuellement une application Java, SIARD Suite, permettant la navigation dans l'archive SIARD et l'ajout ou la modification de métadonnées.

[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
Traduction: Olivier Rouchon, olivier.rouchon@cines.fr