Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Home Affairs FDHA
Swiss Federal Archives SFA
Unit Innovation and Preservation

Hartwig Thomas, 03 October 2013

Document version 1.00

SIARD Suite
Data Type Mapping for DB/2

Published by:

Swiss Federal Archives
Archivstrasse 24
3003 Bern
Switzerland

1 Introduction

The Swiss Federal Archives developed the database archival called SIARD (Software Independent Archiving of Relational Databases) within the framework of the ARELDA (ARchivierung ELektronischer DAten) project. The SIARD format is used for long-term archiving of relational database content.

On behalf of the Swiss Federal Archives, Enter AG implemented the software SIARD Suite which supports converting database content from live proprietary database systems to the normalized SIARD format as well as uploading database content in SIARD format to such a live database system.

Unfortunately most real database systems to not support the SQL:1999 standard fully, on which the SIARD format is based. Therefore SIARD Suite needs to normalize/denormalize the data types during the conversion process.

This document specifies, how the DB/2 data are converted to the SIARD format and how SIARD data are converted to DB/2 on upload.

The conversions are *idempotent*. I.e. after the initial download any number of up- and download can be executed without changing the data types or values.

2 Mapping of SIARD Datatypes

2.1 DB/2 => SIARD

See also

http://publib.boulder.ibm.com/infocenter/db2luw/v10r5/index.jsp?topic=%2Fcom.ibm.db2.luw.sql.ref.doc%2Fdoc%2Fr0008483.html.

DB/2	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
CHAR	CHAR(1)	CHARACTER(1)	xs:string
CHAR(n)	CHAR(n)	CHARACTER(n)	xs:string
VARCHAR(n)	VARCHAR(n)	CHARACTER VARYING(n)	xs:string
LONG VARCHAR	LONGVARCHAR(32700)	CHARACTER LARGE OBJECT	clobType
CLOB	CLOB(1048576)	CHARACTER LARGE OBJECT	clobType
XML	OTHER	XML	clobType
GRAPHIC	CHAR(2)	CHARACTER(1)	xs:string
GRAPHIC(n)	CHAR(2*n)	CHARACTER(n)	xs:string
VARGRAPHIC(n)	VARCHAR(2*n)	CHARACTER(n)	xs:string
DBCLOB	CLOB(2097152)	CHARACTER LARGE OBJECT	clobType
CHAR FOR BIT DATA	BINARY(1)	BIT(8)	xs:hexBinary
CHAR(n) FOR BIT DATA	BINARY(n)	BIT(8*n)	xs:hexBinary
VARCHAR(n) FOR BIT DATA	VARBINARY(n)	BIT VARYING(8*n)	xs:hexBinary
LONG VARCHAR FOR BIT DATA	LONGVARBINARY(32700)	BINARY LARGE OBJECT	blobType
BLOB	BLOB(1048576)	BINARY LARGE OBJECT	blobType

DB/2	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
SMALLINT	SMALLINT(5)	SMALLINT	xs:integer
INTEGER	INTEGER(10)	INTEGER	xs:integer
BIGINT	BIGINT(19)	DECIMAL(19)	xs:decimal
NUMERIC	DECIMAL(5)	DECIMAL(5)	xs:decimal
NUMERIC(p)	DECIMAL(p)	DECIMAL(p)	xs:decimal
NUMERIC(p,s)	DECIMAL(p,s)	DECIMAL(p,s)	xs:decimal
DECIMAL	DECIMAL(5)	DECIMAL(5)	xs:decimal
DECIMAL(p)	DECIMAL(p)	DECIMAL(p)	xs:decimal
DECIMAL(p,s)	DECIMAL(p,s)	DECIMAL(p,s)	xs:decimal
FLOAT	FLOAT(53)	DOUBLE PRECISION	xs:float
FLOAT(p)	p <= 7: REAL(24) p > 7: FLOAT(53)	p <= 7: REAL p > 7: DOUBLE PRECISION	xs:float
REAL	REAL(24)	REAL	xs:float
DOUBLE	DOUBLE(53)	DOUBLE PRECISION	xs:float
DATE	DATE	DATE	xs:date
TIME	TIME	TIME	xs:time
TIMESTAMP	TIMESTAMP(6)	TIMESTAMP	xs:dateTime
TIMESTAMP(n)	TIMESTAMP(n)	TIMESTAMP(n)	xs:dateTime

2.2 SIARD => DB/2

XML	*SQL:1999 (SIARD)*	*DB/2*
xs:decimal	NUMERIC	DECIMAL(31)
xs:decimal	NUMERIC(n)	DECIMAL(n)
xs:decimal	NUMERIC(p,q)	DECIMAL(p,q)

XML	*SQL:1999 (SIARD)*	*DB/2*
xs:decimal	DECIMAL	DECIMAL(31)
xs:decimal	DECIMAL(n)	DECIMAL(n)
xs:decimal	DECIMAL(p,q)	DECIMAL(p,q)
xs:integer	SMALLINT	SMALLINT
xs:integer	INTEGER	INTEGER
xs:integer	BIGINT	BIGINT
xs:float	FLOAT	FLOAT
xs:float	FLOAT(n)	FLOAT(n)
xs:float	REAL	REAL
xs:float	DOUBLE PRECISION	DOUBLE
xs:hexBinary	BIT	CHAR(1) FOR BIT DATA
xs:hexBinary	BIT(n)	n/8 <= 254: CHAR (ceil(n/8)) FOR BIT DATA n/8 > 254 and n/8 <= 32762: VARCHAR(ceil(n/8)) FOR BIT DATA n/8 > 32762: BLOB
xs:hexBinary	BIT(n) VARYING	n/8 > 254 and n/8 <= 32762: VARCHAR(ceil(n/8)) FOR BIT DATA n/8 > 32762: BLOB
xs:hexBinary	BINARY LARGE OBJECT	BLOB
xs:boolean	BOOLEAN	CHAR(1) FOR BIT DATA
xs:string	CHARACTER	CHAR(1)/GRAPHIC(1)
xs:string	CHARACTER(n)	n <= 254/128: CHAR(n)/GRAPHIC(n) n > 254/128 and n <= 32762/16336: VARCHAR(n)/VARGRAPHIC(n) n > 32762/16336: CLOB

XML	*SQL:1999 (SIARD)*	*DB/2*
xs:string	CHARACTER VARYING(n)	n <= 32762/16336: VARCHAR(n)/VARGRAPHIC(n) n > 32762/16336: CLOB
xs:string	CHARACTER LARGE OBJECT	CLOB
xs:string	NATIONAL CHARACTER	CHAR(1)/GRAPHIC(1)
xs:string	NATIONAL CHARACTER(n)	n <= 254/128: CHAR(n)/GRAPHIC(n) n > 254/128 and n <= 32762/16336: VARCHAR(n)/VARGRAPHIC(n) n > 32762/16336: CLOB
xs:string	NATIONAL CHARACTER VARYING(n)	n <= 32762/16336: VARCHAR(n)/VARGRAPHIC(n) n > 32762/16336: CLOB
xs:string	NATIONAL CHARACTER LARGE OBJECT	CLOB
xs:date	DATE	DATE
xs:time	TIME(p)	TIME(8)
xs:dateTime	TIMESTAMP	TIMESTAMP(6)
xs:dateTime	TIMESTAMP(n)	TIMESTAMP(n)

If the DB/2 instance supports graphic characters, then the GRAPHIC types are chosen, in order to preserve UTF-8 characters. If it does not support graphic characters, then one incurs a possible loss of non-ASCII characters.

If a string is longer than 4000 characters then „clobType“ and „xs:string“ are replaced by an external reference to a text file.

If a binary array is longer than 2000 bytes then „blobType“ and „xs:hexBinary“ are replaced by an external reference to a binary file.

Characters that cannot be represented in UNICODE (Codes 0-8, 14-31, 127-159) as well as the escape character '\' and multiple space characters are escaped as \u00<xx> in XML. Less-than and ampersand characters are represented as entity references in XML.