Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Home Affairs FDHA
Swiss Federal Archives SFA

Hartwig Thomas, 4. December 2013

Document version 1.10

SIARD Suite

Data Type Mapping for SQL Server

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 SQL Server data are converted to the SIARD format and how SIARD data are converted to SQL Server 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 SQL Server => SIARD

See also http://msdn.microsoft.com/en-us/library/ms187752(SQL.90).aspx and http://msdn.microsoft.com/en-us/library/ms187752(v=sql.110).aspx.

SQL Server	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
char	CHAR(1)	CHARACTER(1)	xs:string
char(n)	CHAR(n)	CHARACTER(n)	xs:string
varchar	VARCHAR(1)	CHARACTER VARYING(1)	xs:string
varchar(n)	VARCHAR(n)	CHARACTER VARYING(n)	xs:string
text	LONGVARCHAR (2‘147‘483‘647)	CHARACTER LARGE OBJECT	clobType
nchar	OTHER(1)	NATIONAL CHARACTER(1)	xs:string
nchar(n)	OTHER(n)	NATIONAL CHARACTER(n)	xs:string
nvarchar	OTHER(1)	NATIONAL CHARACTER VARYING(1)	xs:string
nvarchar(n)	OTHER(n)	NATTIONAL CHARACTER VARYING(n)	xs:string
ntext	LONGVARCHAR (1‘073‘741‘823)	NATIONAL CHARACTER LARGE OBJECT	clobType
xml	LONGVARCHAR (2‘147‘483‘647)	XML	clobType
tinyint	TINYINT(3)	SMALLINT	xs:integer
smallint	SMALLINT(5)	SMALLINT	xs:integer
int	INTEGER(10)	INTEGER	xs:integer
bigint	BIGINT(19)	NUMERIC(19)	xs:decimal
numeric	NUMERIC(18)	NUMERIC(18)	xs:decimal
numeric(p)	NUMERIC(p)	NUMERIC(p)	xs:decimal
numeric(p,s)	NUMERIC(p,s)	NUMERIC(p,s)	xs:decimal

SQL Server	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
decimal	DECIMAL(18)	DECIMAL(18)	xs:decimal
decimal(p)	DECIMAL(p)	DECIMAL(p)	xs:decimal
decimal(p,s)	DECIMAL(p,s)	DECIMAL(p,s)	xs:decimal
smallmoney	DECIMAL(10,4)	DECIMAL(10,4)	xs:decimal
money	DECIMAL(19,4)	DECIMAL(19,4)	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
bit	BIT(1)	BOOLEAN	xs:boolean
binary	BINARY(1)	BIT(8)	xs:hexBinary
binary(n)	BINARY(n)	BIT(8*n)	xs:hexBinary
varbinary	VARBINARY(1)	BIT VARYING(8)	xs:hexBinary
varbinary(n)	VARBINARY(n)	BIT VARYING(8*n)	xs:hexBinary
image	LONGVARBINARY (2‘147‘483‘647)	BINARY LARGE OBJECT	blobType
sql_variant	LONGVARBINARY	BINARY LARGE OBJECT	blobType
date	DATE	DATE	xs:date
time	TIME	TIME(7)	xs:time
datetime	TIMESTAMP(3)	TIMESTAMP(7)	xs:dateTime
datetime2	TIMESTAMP(7)	TIMESTAMP(7)	xs:dateTime
smalldatetime	TIMESTAMP(0)	TIMESTAMP	xs:dateTime

2.2 SIARD => SQL Server

XML	*SQL:1999 (SIARD)*	*SQL Server*
xs:decimal	NUMERIC	numeric
xs:decimal	NUMERIC(n)	numeric(min(n,28))
xs:decimal	NUMERIC(p,q)	numeric(min(p,28), q-p+min(p,28))
xs:decimal	DECIMAL	decimal
xs:decimal	DECIMAL(n)	decimal(min(n,28))
xs:decimal	DECIMAL(p,q)	decimal(min(p,28), q-p+min(p,28))
xs:integer	SMALLINT	smallint
xs:integer	INTEGER	int
xs:integer	BIGINT	bigint
xs:float	DOUBLE PRECISI-ON	float(15)
xs:float	FLOAT(n)	float(n)
xs:float	REAL	real(7)
xs:hexBinary	BIT	binary
xs:hexBinary	BIT(n)	n/8 <= 8000: binary(ceil(n/8)) n/8 > 8000: image
xs:hexBinary	BIT VARYING(n)	n/8 <= 8000: varbinary(ceil(n/8)) n/8 > 8000: image
xs:hexBinary	BINARY LARGE OB-JECT	image
xs:boolean	BOOLEAN	bit
xs:string	CHARACTER	char
xs:string	CHARACTER(n)	char(n)

XML	*SQL:1999 (SIARD)*	*SQL Server*
xs:string	CHARACTER VARYING(n)	n <= 8000: varchar(n) n > 8000: text
xs:string	CHARACTER LARGE OBJECT	text
xs:string	NATIONAL CHARACTER	nchar
xs:string	NATIONAL CHARACTER(n)	nchar(n)
xs:string	NATIONAL CHARACTER VARYING(n)	n <= 8000: nvarchar(n) n > 8000: ntext
xs:string	NATIONAL CHARACTER LARGE OBJECT	ntext
xs:string	XML	xml
xs:date	DATE	date
xs:time	TIME(p)	time
xs:dateTime	TIMESTAMP	datetime (smalldatetime, if available)
xs:dateTime	TIMESTAMP(p)	datetime (datetime2, if available)

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.