Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

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

Hartwig Thomas, 4. December 2013

Document version 1.10

SIARD Suite
Data Type Mapping for Oracle

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 Oracle data are converted to the SIARD format and how SIARD data are converted to Oracle 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 Oracle => SIARD

See also http://www.cs.umbc.edu/help/oracle8/java.815/a64685/basic3.htm and

http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/sql_elements001.htm#i45441.

In order for character lengths to be respected we set session parameter NLS_LENGTH_SEMANTICS to 'CHAR' (rather than to 'BYTE' which is the unfortunate default!).

Oracle	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
CHAR	CHAR(1)	CHARACTER(1)	xs:string
CHAR(n)	CHAR(n)	CHARACTER(n)	xs:string
VARCHAR2(n)	VARCHAR(n)	CHARACTER VARYING(n)	xs:string
LONG	LONGVARCHAR	CHARACTER LARGE OBJECT	clobType
CLOB	CLOB(4000)	CHARACTER LARGE OBJECT	clobType
NCHAR	OTHER(1)	NATIONAL CHARACTER(1)	xs:string
NCHAR(n)	OTHER(n)	NATIONAL CHARACTER(n)	xs:string
NVARCHAR2(n)	OTHER(n)	NATIONAL CHARACTER VARYING(n)	xs:string
NCLOB	OTHER(4000)	NATIONAL CHARACTER LARGE OBJECT	clobType
XMLTYPE	OTHER(2000)	XML	clobType
INTEGER	DECIMAL(22)	DECIMAL(22)	xs:decimal
NUMBER(n)	DECIMAL(n)	DECIMAL(n)	xs:decimal
NUMBER(p,q)	DECIMAL(p,q)	DECIMAL(p,q)	xs:decimal
FLOAT(p)	FLOAT(p)	FLOAT(p)	xs:float
REAL	FLOAT(63)	FLOAT(63)	xs:float
DOUBLE PRECISION	FLOAT(126)	FLOAT(126)	xs:float

Oracle	*JDBC (java.sql.Types)*	*SQL:1999 (SIARD)*	*XML*
BINARY_FLOAT	100 (oracle.sql.BINARY_FLOAT	FLOAT(24)	xs:float
BINARY_DOUBLE	101 (oracle.sql.BINARY_DOUBLE)	FLOAT(53)	xs:float
RAW(n)	VARBINARY(n)	BIT VARYING(8*n)	xs:hexBinary
LONGRAW	LONGVARBINARY	BINARY LARGE OBJECT	blobType
BLOB	BLOB(4000)	BINARY LARGE OBJECT	blobType
BFILE	-13 (oracle.sql.BFILE)(530)	BINARY LARGE OBJECT	blobType
DATE	TIMESTAMP(7,0)	TIMESTAMP	xs:dateTime
TIMESTAMP(n)	OTHER(7+n,n)	TIMESTAMP(n)	xs:dateTime
TIMESTAMP	TIMESTAMP(6)	TIMESTAMP(6)	xs:dateTime
TIMESTAMP WITH TIME ZONE	oracle.sql.TIMESTAMPTZ(6)	TIMESTAMP(6)	xs:dateTime
TIMESTAMP WITH LOCAL TIME ZONE	oracle.sql.TIMESTAMPLTZ(6)	TIMESTAMP(6)	xs:dateTime
ROWID	OTHER	CHARACTER VARYING	xs:string
UROWID(n)	OTHER	CHARACTER VARYING	xs:string

2.2 SIARD => Oracle

XML	*SQL:1999 (SIARD)*	*Oracle*
xs:decimal	NUMERIC	INTEGER
xs:decimal	NUMERIC(n)	NUMBER(n)
xs:decimal	NUMERIC(p,q)	NUMBER(p,q)

XML	*SQL:1999 (SIARD)*	*Oracle*
xs:decimal	DECIMAL	INTEGER
xs:decimal	DECIMAL(n)	NUMBER(n)
xs:decimal	DECIMAL(p,q)	NUMBER(p,q)
xs:integer	SMALLINT	NUMBER(38)
xs:integer	INTEGER	NUMBER(38)
xs:integer	BIGINT	NUMBER(38)
xs:float	DOUBLE PRECISION	FLOAT(126)
xs:float	FLOAT	FLOAT(126)
xs:float	FLOAT(n)	FLOAT(n)
xs:float	REAL	FLOAT(63)
xs:hexBinary	BIT	RAW(1)
xs:hexBinary	BIT(n)	n/8 <= 4000: RAW(ceil(n/8)) n/8 > 4000: BLOB
xs:hexBinary	BIT VARYING(n)	n/8 <= 4000: RAW(ceil(n/8)) n/8 > 4000: BLOB
xs:hexBinary	BINARY LARGE OBJECT	BLOB
xs:boolean	BOOLEAN	RAW(1)
xs:string	CHARACTER	CHAR
xs:string	CHARACTER(n)	CHAR(n)
xs:string	CHARACTER VARYING(n)	n <= 4000: VARCHAR2(n) n > 4000: CLOB
xs:string	CHARACTER LARGE OBJECT	CLOB
xs:string	NATIONAL CHARACTER	NCHAR
xs:string	NATIONAL CHARACTER(n)	NCHAR(n)
xs:string	NATIONAL CHARACTER VARYING(n)	n <= 4000: NVARCHAR2(n) n > 4000: NCLOB

XML	*SQL:1999 (SIARD)*	*Oracle*
xs:string	NATIONAL CHARACTER LARGE OBJECT	NCLOB
xs:string	XML	XMLTYPE
xs:date	DATE	DATE
xs:time	TIME(p)	DATE
xs:dateTime	TIMESTAMP	DATE
xs:dateTime	TIMESTAMP(p)	TIMESTAMP(p)

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.