

Bern, 2008. szeptember 30.

SIARD

Formátum leírás

Összefoglaló

A dokumentum a Svájci Szövetségi Levéltár SIARD Suite alkalmazása által használt adatformátumot írja le.

SIARD (Software-Independent Archival of Relational Databases) az ARELDA digitális archiválás projekt keretén belül került kifejlesztésre. Ez a dokumentum a relációs adatbázisok hosszú távú megőrzésére szolgáló fájlformátum szabványos leírása.

A SIARD egy nyílt szabvány, melyet a SIARD Suite alkalmazás támogat. Az eszköz segítségével relációs adatbázisokat (mint például MS Access, Oracle és SQL Server) SIARD formátumba lehet konvertálni. A SIARD formátum nyílt szabványokra épül - többek között ISO Unicode, XML, SQL1999 és ZIP szabványokon alapszik. A nemzetközileg elfogadott szabványok alkalmazása biztosítja, a leggyakrabban használt relációs adatbázis modellek hosszú távú megőrzését és hozzáférhetőségét.

Jelenleg a SIARD formátumot a Svájci Szövetségi Levéltár, és több különböző svájci szövetségi hivatal alkalmazza. 2008 májusában, az európai Planets projekt hivatalos relációs adatbázis archiválási formátumaként fogadta el.

Tartalomjegyzék

1	ÁTTEKINTÉS	3
2	A SZABVÁNY HATÓKÖRE	4
3	ALAPELVEK	5
3.1	Szabványok alkalmazása	5
3.2	Adatbázisok, mint dokumentumok	5
3.3	Karakterkészletek és karakterek	5
3.4	Azonosítók	6
4	A SIARD FORMÁTUM	7
4.1	Áttekintés: relációs adatbázisok	7
4.2	A SIARD archív fájl felépítése	7
4.3	Metaadatok egy SIARD archívumban	8
4.3.1	Adatbázis szintű metaadatok	8
4.3.2	Séma szintű metaadatok.....	9
4.3.3	Tábla szintű metaadatok	10
4.3.4	Oszlop szintű metaadatok	10
4.3.5	Elsődleges kulcs metaadatok.....	11
4.3.6	Idegen kulcs metaadatok	11
4.3.7	Hivatkozási metaadatok	11
4.3.8	Jelölőkulcs metaadatok	11
4.3.9	Kényszer metaadatok.....	12
4.3.10	Trigger szintű metaadatok.....	12
4.3.11	Nézet szintű metaadatok.....	12
4.3.12	Eljárás szintű metaadatok	13
4.3.13	Paraméter metaadatok.....	13
4.3.14	Felhasználói szintű metaadatok	13
4.3.15	Szerepkör szintű metaadatok.....	13
4.3.16	Jogosultság szintű metaadatok.....	14
4.4	Elsődleges adatok a SIARD archívumban.....	14
5	FÜGGELÉK: XML SÉMADEFINÍCIÓK	15
5.1	metadata.xsd	15
5.2	metadata.xml	24
5.3	Példa egy tábla XML séma-definíciójára: table0.xsd	29
5.4	Példa egy tábla elsődleges adatára: table0.xml	30

1 Áttekintés

A SIARD formátumleírás műszaki dokumentum, olyan IT szakembereknek szól, akik relációs adatbázisok hosszú távú archiválásával foglalkoznak. A dokumentum a Szövetségi Tanács 2008. január 23-i, a digitális dokumentumok kezelésére vonatkozó határozatán alapszik. A SIARD formátum specifikáció pedig, a 2006. április. 11-i ARELDA projekt archiválási stratégiáján alapul.

A fent említett ARELDA projekt (ARchivierung ELEktronischer DATen) keretén belül, a Svájci Szövetségi Levéltár (BAR)¹ kifejlesztette a SIARD Suite alkalmazást, a relációs adatbázis tartalmak hosszú távú archiválására.

Az alkalmazás eredetileg prototípusként készült, átmenetileg elérhető egy újabb verziója: SIARD Suite néven.

A dokumentum nem az alkalmazást, hanem az adatbázisok archiválására szolgáló formátumot írja le.

A nyílt SIARD adatbázis formátum, a SIARD Suite alkalmazástól függetlenül elektronikus adatbázisok hosszú távú archiválására használható. Ha egy adatbázis szerkezetét és tartalmát SIARD formátumba migráljuk, lehetővé válik a későbbiekben az adatbázis adatainak hozzáférése akkor is, ha az eredeti adatbázis kezelő szoftverek már nem elérhetőek vagy már nem futtathatóak. Ezt úgy érték el, hogy a SIARD formátumban a legmegfelelőbb nemzetközileg támogatott szabványokat alkalmazták. Az adatbázis tartalom hosszú távú értelmezhetősége lényegében két ISO szabványon alapul, az XML és SQL: 1999 szabványokon.

A SIARD formátumot 2008 májusában, az európai Planets² projekt elfogadta, mint a relációs adatbázisok archiválásának hivatalos formátumát.

¹ Bundesarchiv BAR (Svájci Szövetségi Levéltár BAR) [<http://www.bar.admin.ch>]

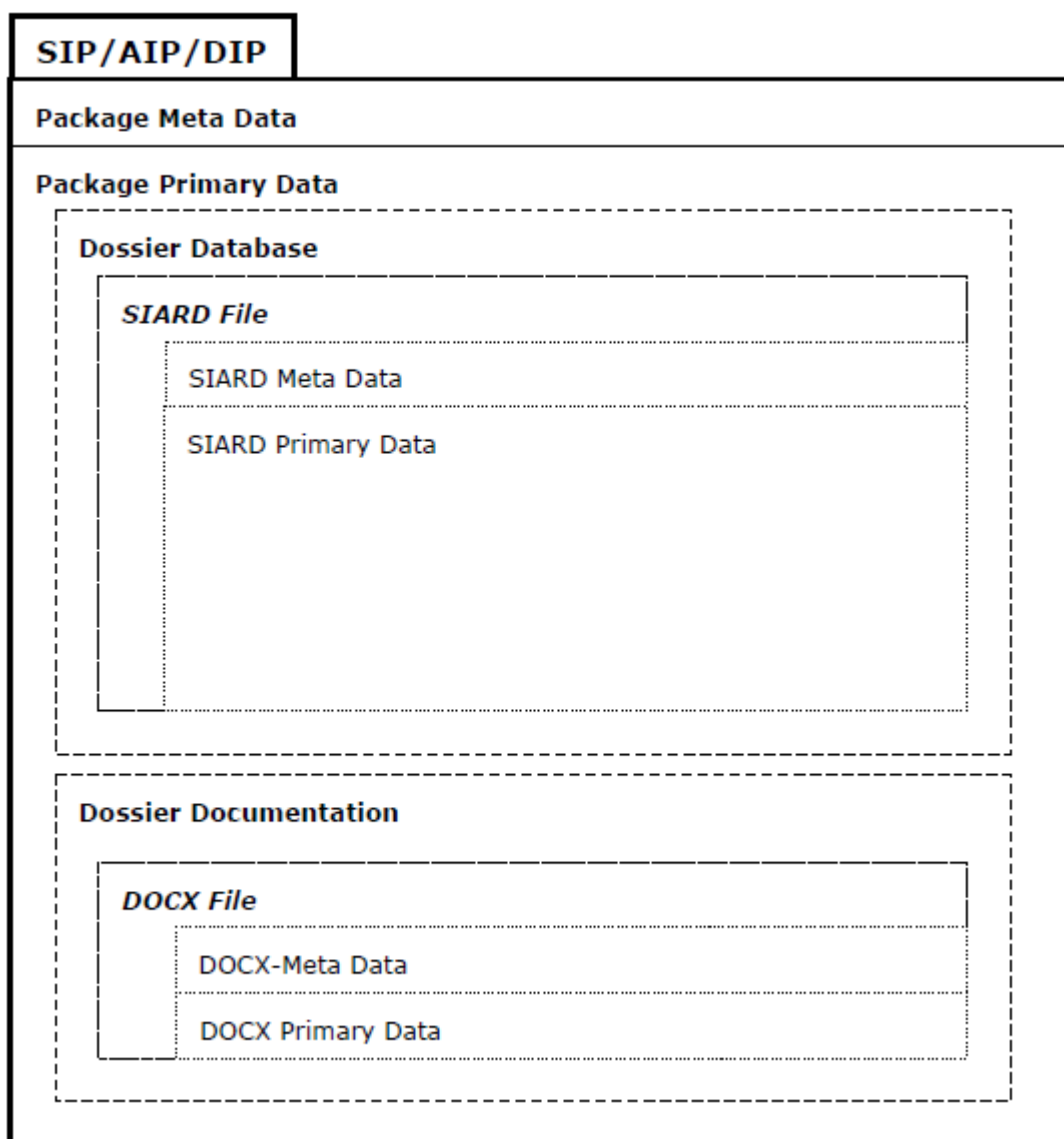
² PLANETS = Preservation and Long-term Access via NETworked Services. Lásd a PLANETS honlapokon: <http://www.planets-project.eu/>

2 A szabvány hatóköre

Meg kell jegyeznünk, hogy a SIARD formátumot, a digitális dokumentumok egy bizonyos fajtájának – nevezetesen, a relációs adatbázisok – hosszú távú tárolására tervezték, teljesen független az OAIS³ modell SIP (Submission Information Package), AIP (Archival Information Package), és a DIP (Dissemination Information Package) csomagjainak felépítésétől.

Egy adatbázis SIARD formátumban archiválható egy archív csomag részeként is, mely tartalmazhat további dokumentumokat is (az adatbázis releváns szakmai tartalmának megértéséhez).

Hasonlóan egy belső metaadat struktúrájval rendelkező modern Word vagy e-mail fájlhoz, mely tartalmazhat elsődleges és különböző kiegészítő adatokat, az archivált relációs adatbázis is tartalmazza az aktuális elsődleges adatok mellett, saját metaadatait, amelyek részletesen leírják az adatbázis rekordokat – függetlenül az OIAS archív csomagban összegyűjtött metaadat katalógustól.



³ OAIS = Offenes Archivinformations-System (Nyílt archívum információs rendszer) (Open Archival Information System), ISO-Standard ISO 14721:2003

3 Alapelvek

3.1 Szabványok alkalmazása

Az adatbázis tartalom hosszú ideig való értelmezhetőségének biztonsága érdekében, a SIARD formátum meghatározóan az XML⁴ és SQL:1999⁵ ISO szabványokon alapszik.

Minden adatbázis tartalmat XML állományokban tárol (séma és SQL kód meghatározások SQL:1999 szabványnak megfelelően). Ezek alól az egyedüli kivételt a nagyobb BLOB és CLOB adatok (Binary Large Objects és Character Large Objects) jelentik, mely adatok tárolása szeparáltan az XML állományban meghivatkozott bináris állományokban történik.

3.2 Adatbázisok, mint dokumentumok

A SIARD formátum koncepciójánál egy relációs adatbázist egyetlen archiválható dokumentumként kezel. Hasonlóan egy Word vagy e-mail fájlhoz, az adatbázis belső szerkezete több részből állhat. Általában a relációs adatbázis minden tábláját együtt kell archiválni, hogy az egyes táblák közti adathivatkozások megmaradjanak.

A fenti analógiát hangsúlyozva, a SIARD formátumban archivált adatbázis is egyetlen fájlban kerül tárolásra. A fájl egyetlen (tömörítetlen) ZIP⁶ archívum, amely egy meghatározott mappastruktúrában tartalmazza a fent említett XML és bináris fájlokat. Különböző adatbázisok külön-külön SIARD fájlba kerülnek.

3.3 Karakterkészletek és karakterek

Általában valamennyi adat Unicode karakterkészletben⁷ van. Olyan adatbázisok esetén, melyek más karakterkészletet támogatnak, az adatkinyerés során megtörténik az Unicode karakterkészletre való konverzió. Ebből kifolyólag, a SIARD az adatbázisban található nemzeti karakterlánc típusú (NCHAR, NVARCHAR és NCLOB) adatokat, átkonvertálja nemzeti típusúvá (CHAR, VARCHAR és/vagy CLOB).

Ez a konvenció az XML által jól támogatott, függetlenül attól, hogy az XML fájl UTF⁸-8 vagy UTF-16 formátumban van.

A SIARD formátumú XML fájlokban bizonyos karakterek, melyeknek az XML szintaxisban speciális jelentésük van, xs:string típusú egyedreferenciával vannak helyettesítve.

Az Unicode 0-31 (kivéve 9, 10 és 13) és 127-159 vezérlő karaktereket a backslash („\”) escape karakter helyettesíti, biztosítva az XML-nek való megfelelést.

Helyettesítő karakterek

Eredeti karakter	Karakter a SIARD formátumban
&	&

⁴ XML = Extensible Markup Language, ISO/IEC 19503:2005

⁵ Jim Melton, Alan R. Simon: SQL:1999 – Understanding Relational Language Components, Morgan Kaufmann Publishers, 2002, ISBN 1-55860-456-1 und

Jim Melton: Advanced SQL:1999 – Understanding Object-Relational and Other Advanced Features, Morgan Kaufmann Publishers, 2003, ISBN 1-55860-677-7

⁶A ZIP fájlok meghatározása eredetileg Phil Katztól ered és manapság nagyon elterjedt. A Microsoft tömörített ZIP mappákat kínál fel Windows operációs rendszer részeként, és a JAVA archívumok (JAR fájlok) is ZIP fájlok.

Az aktuális 6.3.2 verziót, mely a PkWare vállalat publikált specifikáció, megtalálható a következő helyen: <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>.

⁷ Unicode = a karakterkészlet nemzetközi, ISO 10646 szabványoknak megfelelő rendszere, The Unicode Consortium: The Unicode Standard, Reading MA, Addison Wesley, 2000.

⁸ UTF = Unicode Transformation Format (Unicode átalakítási formátum)

Eredeti karakter	Karakter a SIARD formátumban
<	<
0-31	\u0000 - \u001F
127-159	\u007F - \u009F
\	\\

3.4 Azonosítók

Az SQL:1999-ben szóköz és speciális karakterek nélküli reguláris azonosítók találhatók, melyeknél nincs megkülönböztetve a kis és nagybetű, tárolásuk nagybetűvel történik. Az idézőjelbe tett azonosítók tartalmazhatnak speciális karaktereket. A kifejezésekben ezekre az azonosítókra dupla idézőjelben történik a hivatkozás.

Az Unicode szabvány határozza meg, hogy melyek speciális karakterek és melyek az ábécé nagybetűs verziói.

A metaadatokban a reguláris azonosító nagybetűvel, a szóközt tartalmazó azonosító idézőjelben kerül tárolásra. Az SQL:1999 megjegyzi: ha egy azonosító olyan karaktert tartalmaz, amit a reguláris azonosító nem tartalmazhat, körülhatároló azonosítóról beszélünk.

4 A SIARD formátum

4.1 Áttekintés: relációs adatbázisok

Egy **Adatbázis** (Database) rendszerint egy vagy több adatbázis sémából⁹, valamint egyedi felhasználókból, szerepkörökből és az adatbázis bizonyos részeihez hozzáférést biztosító jogok definíciójából áll. Az SQL:1999-ben a **felhasználók** (Users) és **szerepkörök** (Roles) a **privilegiumok** (Privileges) hordozója lehet.

A **sémák** (Schema) a táblák, nézetek és rutinok tárolói.

A **táblákat** (Table) a tábladefiníció határozza meg.

Egy tábladefiníció a következőkből áll:

- ♦ mezőkből – amelyek hozzárendelnek a tábla minden **oszlopához** (Column) egy nevet és egy típust
- ♦ adatrekordokból – amelyek tartalmazzák az aktuális **elsődleges adatokat** (Primary Data)
- ♦ egy opcionális **elsődleges kulcsból** (Primary Key)
- ♦ **idegen kulcsokból** (Foreign Keys) – amelyek biztosítják a hivatkozási integritást
- ♦ **jelölő kulcsokból** (Candidate keys) – melyek biztosítják egy rekord azonosítását a táblában
- ♦ **kényszerekből** (Check Constraint) – melyek biztosítják a konzisztenciát
- ♦ opcionálisan a táblákra lehet triggereket (Triggers) definiálni

A **nézetek** (View) adatbázisban tárolt standard lekérdezések. A lekérdezési eredmény egy tábla, mely mezőket és adatrekordokat tartalmaz.

SQL eljárások (SQL Routine vagy Stored Procedure) - elsődlegesen a nézetek/lekérdezések megértésénél fontosak, ahol a rész kifejezésekben előfordulhatnak.

A relációs adatbázis tehát több strukturált adatbázis objektumból (pl. séma, nézet, stb.), valamint tábla adatokból áll össze.

4.2 A SIARD archív fájl felépítése

A SIARD formátumban archivált relációs adatbázis két komponensből tevődik össze: metaadatokból, melyek az archivált adatbázis struktúráját írják le, és elsődleges adatokból, melyek a táblák tartalmát hordozzák. A metaadatok nyújtanak információt arról, hogy melyik elsődleges adat hol található az archív állományban.

Az adatbázis meta- és elsődleges adatai, egy tömörítetlen ZIP formátumú fájlban vannak. Az archív állomány fájlnevének kiterjesztése „.siard”. A metaadatok a *header*, míg az elsődleges adatok a *content* könyvtárban kerülnek.

A következő ábrázolás ezt a felépítést szemlélteti:

header

metadata.xsd¹⁰

metadata.xml

content

schema1

table1

table.xsd

⁹ Egy adatbázis séma egyfajta névspace- prefix. Egy adatbázis katalógus minden séma metaadatot tartalmaz a katalógusban. A katalógus szint SQL:1999ben megegyezik a „dokumentum adatbázissal” mely SIARD formátumban egy archív formátummá változhat át.

¹⁰ XSD = XML séma meghatározás

```

table.xml

lob111
    record1.txt / record1.bin
lob2
    record1.txt / record1.bin
...
table2
    table.xsd
    table.xml
...
schema2
...

```

A következő leírások a két mappa formátumát és a bennük levő fájlokat részletezik.

4.3 Metaadatok egy SIARD archívumban

A SIARD archívum metaadat mappája tárolja az archivált adatbázis struktúráját és megmutatja, melyik elsődleges adat hol van az archívum állományban.

Minden metaadat egyetlen metadata.xml fájlban van összegyűjtve a header könyvtárban. A fájl a relációs adatbázishoz hasonló hierarchia szerint épül fel.

A következő fejezetek leírják az egyéni szintek tartalmát. Az opcionálitást jelölő – „opc.” – oszlopban szereplő „igen” jelöli azt, hogy az elem opcionális, vagyis hiánya megengedett.

4.3.1 Adatbázis szintű metaadatok

A *metadata.xml* fájl a következő globális információkat tartalmazza az adatbázis szinten:

Azonosító	opc.	Leírás
version	nem	A formátum verzió (SIARD formátum verzió, nem egy SIARD programé) jelenleg mindig „1.0“
dbname	nem	Adatbázis rövid azonosítója
description	igen	Az egész adatbázis értelmezésének és tartalmának leírása
archiver	igen	Annak a személynek a neve, aki az adatbázis elsődleges adatainak archiválását végezte
archiverContact	igen	Annak a személynek az elérhetőségi adatai (telefon, e-mail) aki az adatbázis elsődleges adatainak archiválását végezte
dataOwner	nem	Az adatbázis adatainak tulajdonosa; intézmény vagy személy, aki az archiválás pillanatában birtokolta az adatok hozzáféréséhez szükséges engedélyeket, továbbá aki a jogi előírásokért és adatvédelemért felelős.
dataOriginTimespan	nem	Az adatbázis adatainak keletkezési időszaka, egy hozzávetőleges idői, szövegesen

¹¹ LOB = Large Objects etwa BLOB = Binary Large Objects und CLOB = Character Large Objects, lásd a 4.1-es bekezdésben

Azonosító	opc.	Leírás
producerApplication	igen	Az alkalmazás neve és verziója, amely eredetileg létrehozta a SIARD fájlt az adatbázisból.
archivalDate	nem	Archiválás dátuma; a elsődleges adatok archiválásának dátuma
messageDigest	nem	A /content könyvtárra kiszámolt hexadecimális üzenet kivonat (Hexadecimal message digest) kódja, ahol a prefix megmutatja a kivonatkód számítási algoritmusának (Digest-Algorithm) típusát (MD512 vagy SHA113). Az üzenet kivonat lehetővé teszi az elsődleges adatok integritásának ellenőrzését.
clientMachine	igen	DNS ¹⁴ a számítógép (client) neve, melyen az archiválást végezték
databaseProduct	igen	Adatbázis termék és verzió, amelyből az elsődleges adat archiválása történt
connection	igen	Az elsődleges adatok archiválásához használt kapcsolódási információ
databaseUser	igen	A SIARD alkalmazás felhasználójának a neve, mellyel kapcsolódik az adatbázishoz és archiválja az elsődleges adatokat
schemas	nem	Adatbázis sémák listája
Users	nem	Adatbázis felhasználók listája
roles	igen	Adatbázis szerepekörök listája
Privileges	igen	Felhasználó és szerepkör privilégiumok listája

4.3.2 Séma szintű metaadatok

A séma metaadatokat a metadata.xml fájl tartalmazza, az adatbázis globális információival együtt.

A *metadata.xml* fájl a következő, a sémára vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	Nem	Adatbázisséma név
folder	Nem	A séma könyvtár neve a <i>content</i> alatt a SIARD archívumban
description	Igen	Séma szerepének és tartalmának leírása
tables	Nem	Adatbázis táblák listája
views	igen	Az adatbázisban tárolt lekérdezések listája
routines	igen	Séma rutinok (korábban tárolt eljárások) listája

Minden egyes séma elsődleges adata a SIARD archívum content alkönyvtárába kerül archiválásra. Az adatbázis egy sémájának nevét nem lehet egyben könyvtárnévként is használni, mert az SQL megenged olyan sémanevet, amelyek több fájlrendszerben nem megengedettek.

¹² MD5 = Message-Digest Algoritmus 5, (Message-Digest Algorithm 5)

¹³ SHA = Secure Hash Algorithm

¹⁴ DNS = (Domain Name System) Tartománynév rendszer, egy szétosztott adatbázis, mely a névhelyet az Internethez irányítja

A SIARD formátumon belül csak a legegyszerűbb könyvtár és fájlnevek elfogadhatóak, amelyek ASCII-jeleket engedélyeznek (pl. számok és betűk speciális karakterek nélkül), vagyis amelyek garantáltan minden fájlrendszerrel kompatibilisek.

A SIARD ezért a sémáknak a következő mappaneveket hozza létre schema1, schema2, ...

4.3.3 Tábla szintű metaadatok

A tábla szintű metaadatokat ugyanúgy, mint a globális információkat és séma-metaadatokat a metadata.xml fájl tárolja.

A metadata.xml fájl a következő, a táblára vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	A tábla neve a sémában
folder	nem	A táblát tartalmazó könyvtár neve a séma mappában
description	igen	A tábla szerepének és tartalmának leírása
columns	nem	A tábla oszlopainak listája
primaryKey	igen	A tábla elsődleges kulcsa
foreignKeys	igen	A tábla idegen kulcsainak listája
candidateKeys	igen	A tábla jelölő kulcsainak listája
checkConstraints	igen	A táblára létrehozott kényszerek listája
triggers	igen	A táblán működő triggerek listája
rows	nem	Datasetek száma

Egy tábla elsődleges adatai a SIARD archívumban a content mappa egyik séma alkönyvtárába kerülnek, oda, ahova a tábla tartozik. A SIARD automatikusan létrehozza a table1, table2, table3...., neveket.

4.3.4 Oszlop szintű metaadatok

Az oszlop szintű metaadatokat ugyanúgy, mint a globális információkat, a séma metaadatokat és a tábla szintű metaadatokat a metadata.xml fájl tárolja.

A metadata.xml fájl a következő, az oszlopra vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	A tábla oszlopnevei
folder	igen	Az LOB-mappa neve a tábla mappában
type	nem	Az SQL:1999 szerinti oszloptípus
typeOriginal	igen	Eredeti oszloptípus
defaultValue	igen	Alapértelmezett érték
nullable	igen	Nem kötelező értéket megadni
description	igen	Az oszlop szerepe és tartalma

Mivel a magukat SQL konformnak nevező adatbázis programok nagyon eltérő adattípusokat támogatnak, az SQL:1999 típus mellett az eredeti típus is rögzítésre kerül. Ezért a megfelelő SQL: 1999 típusra történő konvertálást definiálni és dokumentálni kell minden SIARD által támogatott adatbáziskezelőre.

A választható LOB nevű mappa csak a Large Object típusú oszlopok esetén (pl. BLOB vagy CLOB) szükséges. A SIARD automatikusan létrehozza a lob1, lob2, ... neveket.

A Large Object típusú adatmezők tartalma record1.txt, record2.txt, vagy record1.bin, record2.bin ... névvel, ebben a könyvtárban kerülnek létrehozásra. Ezekre a fájlokra az adatokat tartalmazó XML fájlokból történik a hivatkozás.

A nullable érték kivételével valamennyi oszlopra vonatkozó kényszer a táblára vonatkozó kényszerként kerül meghatározásra.

4.3.5 Elsődleges kulcs metaadatok

A *metadata.xml* fájl a következő, az elsődleges kulcsra vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	Elsődleges kulcs neve
column	nem	Elsődleges kulcsot alkotó oszlopnevek listája
description	igen	Elsődleges kulcs szerepe és tartalma

4.3.6 Idegen kulcs metaadatok

A *metadata.xml* fájl a következő, az idegen kulcsra vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	Idegen kulcs neve
referenced-Schema	nem	Az idegenkulcs által hivatkozott tábla sémájának neve
referencedTable	nem	Az idegenkulcs által hivatkozott tábla neve
reference	nem	Hivatkozás (oszlopok és hivatkozott oszlopok listája)
matchType	igen	Az illeszkedés típusa (FULL, PARTIAL vagy SIMPLE) (teljes, részleges vagy egyszerű)
deleteAction	igen	Törlés műveletek, pl.: CASCADE
updateAction	igen	Módosítási műveletek, pl.: SET DEFAULT
description	igen	Az idegen kulcs szerepe és tartalma

A külső táblanevek hivatkozása a *táblanév* vagy *séma.táblanév* módon történhet. Ilyenkor a tagolt azonosítók idézőjelbe kerülnek.

Az illeszkedés típusa lehet FULL, PARTIAL vagy SIMPLE. A törlés és a módosítás, az SQL:1999 szabványban jóváhagyott műveleteket tartalmazza.

4.3.7 Hivatkozási metaadatok

A hivatkozási lista a következő elemekből áll:

Azonosító	opc.	Leírás
column	nem	Oszlop neve
referenced	nem	Hivakozott oszlop neve

4.3.8 Jelölőkulcs metaadatok

A *metadata.xml* fájl a következő, a jelölőkulcsra vonatkozó metaadatokat tartalmazza:

<i>Azonosító</i>	<i>opc.</i>	<i>Leírás</i>
name	nem	Jelölőkulcs neve az adatbázis sémában
column	nem	A jelölőkulcsot alkotó oszlopok neveinek listája
description	nem	A jelölőkulcs szerepe és tartalma

4.3.9 Kényszer metaadatok

A kényszer (megszorítás) egy feltétel ellenőrzésből áll. Ez az SQL:1999 szintaxisa szerint egy Boolean típusú kifejezés true, false vagy unknow (igaz, hamis vagy ismeretlen) értékkel.

A *metadata.xml* fájl a következő, a kényszerekre vonatkozó metaadatokat tartalmazza:

<i>Azonosító</i>	<i>opc.</i>	<i>Leírás</i>
name	nem	Ellenőrző megszorítás neve
condition	nem	Ellenőrző megszorítás feltételek
description	igen	Ellenőrző megszorítás szerepe és tartalma

4.3.10 Trigger szintű metaadatok

A *metadata.xml* fájl a következő, a triggerekre vonatkozó metaadatokat tartalmazza:

<i>Azonosító</i>	<i>opc.</i>	<i>Leírás</i>
name	nem	Trigger neve a táblában
actionTime	nem	BEFORE vagy AFTER
triggerEvent	nem	INSERT, DELETE, UPDATE [OF <trigger column list>]
aliasList	igen	<old or new value alias list>
triggeredAction	nem	<triggered action>
description	igen	A triggerek szerepe és tartalma

4.3.11 Nézet szintű metaadatok

A *metadata.xml* fájl a következő, a nézetre vonatkozó metaadatokat tartalmazza:

<i>Azonosító</i>	<i>opc.</i>	<i>Leírás</i>
name	nem	A nézet neve a sémában
columns	nem	A nézetben szereplő oszlopok neveinek listája
query	igen	Az SQL:1999 szerinti lekérdezés, amely a nézetet meghatározza
queryOriginal	igen	Eredeti SQL- lekérdezés, mely a nézetet meghatározza
description	igen	A nézet szerepe és tartalma

Mivel a magukat SQL konformnak nevező adatbáziskezelő programok nagyon eltérő szintaxis szerint megírt lekérdezéseket engedélyeznek, az SQL:1999 lekérdezés mellett az eredeti lekérdezés is

megőrzésre kerül. A SIARD formátum által támogatott adatbáziskezelők saját szintaxisában megírt lekérdezéseinek fordítását az SQL:1999 szerinti lekérdezésekre definiálni és dokumentálni kell. A nézetek oszlopainak a metaadatszerkezete megegyezik a tábla oszlopainak metadatszerkezetével.

4.3.12 Eljárás szintű metaadatok

A *metadata.xml* fájl a következő, az eljárásokra vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	Az eljárás neve a sémában
description	igen	Az eljárás szerepe és tartalma
source	igen	Az eljárás eredeti forráskódja (VBA, PL/SQL, JAVA)
body	igen	SQL:1999 szerinti rutin forráskód
characteristic	igen	Az eljárás jellemzői
returnType	igen	Az eljárás visszatérési típusa (ha függvény)
parameters	igen	Paraméter lista

Mivel sok adatbáziskezelő program rendelkezik saját eljárásokkal, amelyeket nem lehet megfelelő, SQL:1999 – kompatibilis lekérdezésekre fordítani, az eljárás eredeti forráskódját is lehet archiválni (pl. PL/SQL Oracle adatbázisnál, vagy VBA MS Access Modulnál)

4.3.13 Paraméter metaadatok

A *metadata.xml* fájl a következő paraméterekre vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	Paraméter neve
mode	nem	Paraméter mód (IN, OUT vagy INOUT)
type	nem	Paraméter SQL:1999- típusa
typeOriginal	igen	Eredeti paramétertípus
description	igen	Az eljárás szerepe és jelentése

Ugyanúgy, mint az oszlopok leírása esetén, az eredeti, saját paramétertípust meg lehet adni és tárolni lehet.

4.3.14 Felhasználói szintű metaadatok

A *metadata.xml* fájl a következő, a felhasználóra (User) vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	Nem	Felhasználó neve
description	Igen	A felhasználó szerepe és szignifikanciája/fontossága (milyen „erős” jogokkal rendelkezik)

4.3.15 Szerepkör szintű metaadatok

A *metadata.xml* fájl a következő, a szerepkörre vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
name	nem	Szerepkör neve
admin	nem	Adminisztrációs szerepkör vagy sem
description	igen	A szerepkör szerepe és leírása

4.3.16 Jogosultság szintű metaadatok

A *metadata.xml* fájl a következő, a jogosultságra vonatkozó metaadatokat tartalmazza:

Azonosító	opc.	Leírás
type	nem	Biztosított (engedélyezett) jogosultság (pl. SELECT)
object	igen	Objektum, mely a kiváltságot alkalmazza
grantor	nem	A jogosultság kiosztója
grantee	nem	A jogosultság befogadója (felhasználó vagy szerepkör)
option	igen	Jogosultság engedélyezési opció (ADMIN vagy GRANT)
description	Igen	Jogosultság szerepe és leírása

4.4 Elsődleges adatok a SIARD archívumban

Ahogy a leírtakban olvashattuk, a relációs adatbázis elsődleges adatai a SIARD archívum root könyvtárában lévő content könyvtárban találhatóak. Ha ez a mappa üres (és a metaadatban nincs jelen ellenőrző kód), az egy üres SIARD archív állomány, amely adatbázis struktúrát leíró metaadat definíciókat tartalmaz.

Az egyes táblák metaadatai a SIARD archív állomány content könyvtárában kerülnek archiválásra egy séma alkönyvtárban, amely séma a táblát tartalmazza. A SIARD automatikusan generálja a séma1, séma2, séma3,... neveket a sémákhoz és a tábla1. tábla2, tábla3,...neveket a táblákhoz.

A táblák elsődleges adatai egy table.xml nevű XML fájlba kerülnek tárolásra.

Táblánként egy XML séma definíció (table.xsd) készül, amely meghatározza az elsődleges adat XML tárolási formátumát. Ez a sémadefiníció visszatükrözi a táblázat SQL séma metaadatait és meghatározza, hogy a tábla sorok sorozataként van tárolva, ahol a sorok különböző XML-típusú oszlopbejegyzések sorozatát tartalmazzák. A „tábla” címke/rész neve „table”, a „rekord” címke neve „row”, az oszlop címke neve c1, c2... (pl. siard:c1 ..., amennyiben az XSD névtér (Namespace) is ki van írva).

Az értékek az SQL/XML¹⁵ szerint, az XML típusnak megfelelően kerülnek átalakításra.

Amennyiben egy ilyen adattábla nagy objektum típusokat (LOB, Large Object) (BLOB, CLOB) tartalmaz, amelyek nagyobbak, mint 4000 karakter, vagy/és 2000 Byte, akkor mindegyik mező számára külön fájl lesz létrehozva. A SIARD automatikusan hozza létre a lob1, lob2 mappaneveket minden érintett oszlophoz. A fájlok, amelyek a nagy objektum mezőket képviselik, ebbe a mappába lesznek létrehozva és record1.txt, record2.txt ill. record1.bin, record2.bin, nevet kapnak.

Hogy elkerüljük az üres mappák létrehozását, a mappákat csak szükség esetén kell létrehozni, tehát akkor, ha adatokat tartalmaznak.

¹⁵ <http://www.sqlx.org/>

5 Függelék: XML sémadefiníciók

5.1 metadata.xsd

Az XML séma definíció (metadata.xsd) meghatározza a metadata.xml fájl struktúráját a header könyvtárban.

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- $Workfile: metadata.xsd $ ***** -->
<!-- Metadata schema for SIARD 1.0 -->
<!-- Version : $Id: metadata.xsd 680 2008-02-14 19:58:41Z hartwig $ -->
<!-- Application: SIARD Suite -->
<!-- Software-Independent Archival of Relational Databases -->
<!-- Platform : XML 1.0, XML Schema 2001 -->
<!-- Description: This XML schema definition defines the structure -->
<!-- of the metadata in the SIARD format -->
<!-- ***** -->
<!-- Copyright : 2007, Swiss Federal Archives, Berne, Switzerland -->
<!-- ***** -->
<xs:schema id="metadata"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd"
  targetNamespace="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- root element of an XML file conforming to this XML schema -->
  <xs:element name="siardArchive">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>
          Root element of meta data of the SIARD archive
        </xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <!-- name of the archived database -->
        <xs:element name="dbname" type="mandatoryString"/>
        <!-- short free form description of the database content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>
        <!-- name of person responsible for archiving the database -->
        <xs:element name="archiver" type="xs:string" minOccurs="0"/>
        <!-- contact data (telephone number or email adress) of archiver -->
        <xs:element name="archiverContact" type="xs:string" minOccurs="0"/>
        <!-- name of data owner (section and institution responsible for data)
        of database when it was archived -->
        <xs:element name="dataOwner" type="mandatoryString"/>
        <!-- time span during which data where entered into the database -->
        <xs:element name="dataOriginTimespan" type="mandatoryString"/>
        <!-- date of creation of archive (automatically generated by SIARD) -->
        <xs:element name="archivalDate" type="xs:date"/>
        <!-- message digest code over all primary data in folder "content" -->
        <xs:element name="messageDigest" type="digestType"/>
        <!-- DNS name of client machine from which SIARD was running for archiving -->
        <xs:element name="clientMachine" type="xs:string" minOccurs="0"/>
        <!-- name of database product and version from which database originates -->
        <xs:element name="databaseProduct" type="xs:string" minOccurs="0"/>
        <!-- connection string used for archiving -->
        <xs:element name="connection" type="xs:string" minOccurs="0"/>
        <!-- database user used for archiving -->
        <xs:element name="databaseUser" type="xs:string" minOccurs="0"/>
        <!-- list of schemas in database -->
        <xs:element name="schemas" type="schemasType"/>
        <!-- list of users in the archived database -->
        <xs:element name="users" type="usersType"/>
        <!-- list of roles in the archived database -->
        <xs:element name="roles" type="rolesType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <!-- list of privileges in the archived database -->
        <xs:element name="privileges" type="privilegesType" minOccurs="0"/>
    </xs:sequence>
    <!-- constraint: version number must be 1.0 -->
    <xs:attribute name="version" type="versionType" use="required" />
</xs:complexType>
</xs:element>

<!-- complex type schemas -->
<xs:complexType name="schemasType">
    <xs:annotation>
        <xs:documentation>
            List of schemas
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="schema" type="schemaType" minOccurs="1" maxOccurs="unbounded"
        />
    </xs:sequence>
</xs:complexType>

<!-- complex type schema -->
<xs:complexType name="schemaType">
    <xs:annotation>
        <xs:documentation>
            Schema element in siardArchive
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <!-- database name of the schema -->
        <xs:element name="name" type="xs:string" />
        <!-- archive name of the schema folder -->
        <xs:element name="folder" type="fsName"/>
        <!-- description of the schema's meaning and content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>
        <!-- list of tables in the schema -->
        <xs:element name="tables" type="tablesType"/>
        <!-- list of views in the schema -->
        <xs:element name="views" type="viewsType" minOccurs="0"/>
        <!-- list of routines in the archived database -->
        <xs:element name="routines" type="routinesType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<!-- complex type tables -->
<xs:complexType name="tablesType">
    <xs:annotation>
        <xs:documentation>
            List of tables
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="table" type="tableType" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!-- complex type table -->
<xs:complexType name="tableType">
    <xs:annotation>
        <xs:documentation>
            Table element in siardArchive
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <!-- database name of the table -->
        <xs:element name="name" type="xs:string"/>
        <!-- archive name of the table folder -->
        <xs:element name="folder" type="fsName"/>
        <!-- description of the table's meaning and content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>

```



```

    <!-- list of columns of the table -->
    <xs:element name="columns" type="columnsType"/>
    <!-- primary key -->
    <xs:element name="primaryKey" type="primaryKeyType" minOccurs="0"/>
    <!-- foreign keys -->
    <xs:element name="foreignKeys" type="foreignKeysType" minOccurs="0"/>
    <!-- candidate keys (unique constraints) -->
    <xs:element name="candidateKeys" type="candidateKeysType" minOccurs="0"/>
    <!-- list of (check) constraints -->
    <xs:element name="checkConstraints" type="checkConstraintsType" minOccurs="0"/>
    <!-- list of triggers -->
    <xs:element name="triggers" type="triggersType" minOccurs="0"/>
    <!-- number of rows in the table -->
    <xs:element name="rows" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type views -->
<xs:complexType name="viewsType">
  <xs:annotation>
    <xs:documentation>
      List of views
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="view" type="viewType" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type view -->
<xs:complexType name="viewType">
  <xs:annotation>
    <xs:documentation>
      View element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the view -->
    <xs:element name="name" type="xs:string" />
    <!-- SQL query string defining the view -->
    <xs:element name="query" type="xs:string" minOccurs="0"/>
    <!-- original query string defining the view -->
    <xs:element name="queryOriginal" type="xs:string" minOccurs="0"/>
    <!-- description of the view's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <!-- list of columns of the view -->
    <xs:element name="columns" type="columnsType"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type columns -->
<xs:complexType name="columnsType">
  <xs:annotation>
    <xs:documentation>
      List of columns
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="column" type="columnType" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type column -->
<xs:complexType name="columnType">
  <xs:annotation>
    <xs:documentation>
      Column element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>

```

```

    <!-- database name of the column -->
    <xs:element name="name" type="xs:string" />
    <!-- archive name of the lob folder -->
    <xs:element name="folder" type="fsName" minOccurs="0"/>
    <!-- SQL:1999 data type of the column -->
    <xs:element name="type" type="xs:string" />
    <!-- original data type of the column -->
    <xs:element name="typeOriginal" type="xs:string" minOccurs="0"/>
    <!-- default value -->
    <xs:element name="defaultValue" type="xs:string" minOccurs="0"/>
    <!-- nullability -->
    <xs:element name="nullable" type="xs:boolean"/>
    <!-- unique, references, check column constraints
    are stored as table constraints -->
    <!-- description of the column's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type primaryKey -->
<xs:complexType name="primaryKeyType">
  <xs:annotation>
    <xs:documentation>
      primaryKey element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the primary key -->
    <xs:element name="name" type="xs:string" minOccurs="0" />
    <!-- description of the primary key's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <!-- columns belonging to the primary key -->
    <xs:element name="column" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type foreignKeys -->
<xs:complexType name="foreignKeysType">
  <xs:annotation>
    <xs:documentation>
      List of foreign key constraints
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="foreignKey" type="foreignKeyType" minOccurs="1"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type foreignKey -->
<xs:complexType name="foreignKeyType">
  <xs:annotation>
    <xs:documentation>
      foreignKey element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the foreign key -->
    <xs:element name="name" type="xs:string" />
    <!-- referenced schema -->
    <xs:element name="referencedSchema" type="xs:string"/>
    <!-- referenced table -->
    <xs:element name="referencedTable" type="xs:string"/>
    <!-- references -->
    <xs:element name="reference" type="referenceType" minOccurs="1"
      maxOccurs="unbounded"/>
    <!-- match type (FULL, PARTIAL, SIMPLE) -->
    <xs:element name="matchType" type="matchTypeType" minOccurs="0"/>
    <!-- ON DELETE action e.g. ON DELETE CASCADE -->
    <xs:element name="deleteAction" type="xs:string" minOccurs="0"/>
    <!-- ON UPDATE action e.g. ON UPDATE SET DEFAULT -->

```

```

    <xs:element name="updateAction" type="xs:string" minOccurs="0"/>
    <!-- description of the foreign key's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type reference -->
<xs:complexType name="referenceType">
  <xs:annotation>
    <xs:documentation>
      reference element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- referencing column -->
    <xs:element name="column" type="xs:string"/>
    <!-- referenced column (table.column) -->
    <xs:element name="referenced" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type candidateKeys -->
<xs:complexType name="candidateKeysType">
  <xs:annotation>
    <xs:documentation>
      List of candidate key (unique) constraints
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="candidateKey" type="candidateKeyType" minOccurs="1" maxOc-
      curs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type candidateKey -->
<xs:complexType name="candidateKeyType">
  <xs:annotation>
    <xs:documentation>
      candiate key (unique) element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the candidate key -->
    <xs:element name="name" type="xs:string"/>
    <!-- description of the candidate key's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <!-- columns belonging to the candidate key -->
    <xs:element name="column" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type check constraints -->
<xs:complexType name="checkConstraintsType">
  <xs:annotation>
    <xs:documentation>
      List of check constraints
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="checkConstraint" type="checkConstraintType" minOccurs="1"
      maxOc-curs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type check constraint -->
<xs:complexType name="checkConstraintType">
  <xs:annotation>
    <xs:documentation>
      Check constraint element in siardArchive

```

```

    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the constraint -->
    <xs:element name="name" type="xs:string"/>
    <!-- check condition -->
    <xs:element name="condition" type="xs:string"/>
    <!-- description of the constraint's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type triggers -->
<xs:complexType name="triggersType">
  <xs:annotation>
    <xs:documentation>
      List of triggers
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="trigger" type="triggerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type trigger -->
<xs:complexType name="triggerType">
  <xs:annotation>
    <xs:documentation>
      Trigger element in siardArchive
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the trigger -->
    <xs:element name="name" type="xs:string" />
    <!-- action time -->
    <xs:element name="actionTime" type="actionTimeType"/>
    <!-- trigger event INSERT, DELETE, UPDATE [OF <trigger column list>] -->
    <xs:element name="triggerEvent" type="xs:string"/>
    <!-- alias list <old or new values alias> -->
    <xs:element name="aliasList" type="xs:string" minOccurs="0"/>
    <!-- triggered action -->
    <xs:element name="triggeredAction" type="xs:string"/>
    <!-- description of the trigger's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type routines -->
<xs:complexType name="routinesType">
  <xs:annotation>
    <xs:documentation>
      List of routines
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="routine" type="routineType" minOccurs="1"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type routine -->
<xs:complexType name="routineType">
  <xs:annotation>
    <xs:documentation>
      Routine
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of routine in schema -->

```

```

    <xs:element name="name" type="xs:string"/>
    <!-- description of the routines's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <!-- original source code (VBA, PL/SQL, ...) defining the routine -->
    <xs:element name="source" type="xs:string" minOccurs="0"/>
    <!-- SQL:1999 body of routine -->
    <xs:element name="body" type="xs:string" minOccurs="0"/>
    <!-- routine characteristic -->
    <xs:element name="characteristic" type="xs:string" minOccurs="0"/>
    <!-- SQL:1999 data type of the return value (for functions) -->
    <xs:element name="returnType" type="xs:string" minOccurs="0"/>
    <!-- list of parameters -->
    <xs:element name="parameters" type="parametersType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type parameters -->
<xs:complexType name="parametersType">
  <xs:annotation>
    <xs:documentation>
      List of parameters of a routine
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="parameter" type="parameterType" minOccurs="1"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type parameter -->
<xs:complexType name="parameterType">
  <xs:annotation>
    <xs:documentation>
      Parameter of a routine
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- name of parameter -->
    <xs:element name="name" type="xs:string"/>
    <!-- mode of parameter (IN, OUT, INOUT) -->
    <xs:element name="mode" type="xs:string"/>
    <!-- SQL:1999 type of argument -->
    <xs:element name="type" type="xs:string"/>
    <!-- original data type of the argument -->
    <xs:element name="typeOriginal" type="xs:string" minOccurs="0"/>
    <!-- description of the parameter's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!-- complex type users -->
<xs:complexType name="usersType">
  <xs:annotation>
    <xs:documentation>
      List of users
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="user" type="userType" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!-- complex type user -->
<xs:complexType name="userType">
  <xs:annotation>
    <xs:documentation>
      User
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- user name -->

```

```

        <xs:element name="name" type="xs:string"/>
        <!-- description of the user's meaning and content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<!-- complex type roles -->
<xs:complexType name="rolesType">
    <xs:annotation>
        <xs:documentation>
            List of roles
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="role" type="roleType" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!-- complex type role -->
<xs:complexType name="roleType">
    <xs:annotation>
        <xs:documentation>
            Role
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <!-- role name -->
        <xs:element name="name" type="xs:string"/>
        <!-- role ADMIN (user or role) -->
        <xs:element name="admin" type="xs:string"/>
        <!-- description of the role's meaning and content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<!-- complex type privileges -->
<xs:complexType name="privilegesType">
    <xs:annotation>
        <xs:documentation>
            List of grants
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="privilege" type="privilegeType" minOccurs="1"
            maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!-- complex type privilege -->
<xs:complexType name="privilegeType">
    <xs:annotation>
        <xs:documentation>
            Grant (incl. grant of role)
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <!-- privilege type (incl. ROLE privilege or "ALL PRIVILEGES" -->
        <xs:element name="type" type="xs:string"/>
        <!-- privilege object (may be omitted for ROLE privilege) -->
        <xs:element name="object" type="xs:string" minOccurs="0"/>
        <!-- GRANTED BY -->
        <xs:element name="grantor" type="xs:string"/>
        <!-- user list of users or roles or single value "PUBLIC" -->
        <xs:element name="grantee" type="xs:string"/>
        <!-- optional option "GRANT" or "ADMIN" -->
        <xs:element name="option" type="privOptionType" minOccurs="0"/>
        <!-- description of the grant's meaning and content -->
        <xs:element name="description" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

```

```

<!-- simple type for digest string -->
<xs:simpleType name="digestType">
  <xs:annotation>
    <xs:documentation>
      digestType must be empty or prefixed by MD5 oder SHA1
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:pattern value="(|(MD5|SHA-1).*)" />
  </xs:restriction>
</xs:simpleType>

<!-- simple type for version number -->
<xs:simpleType name="versionType">
  <xs:annotation>
    <xs:documentation>
      versionType must be constrained to "1.0"
      for conformity with this XLM schema
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:enumeration value="1.0"/>
  </xs:restriction>
</xs:simpleType>

<!-- simple type for privilege option -->
<xs:simpleType name="privOptionType">
  <xs:annotation>
    <xs:documentation>
      privOptionType must be "ADMIN" or "GRANT"
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:enumeration value="ADMIN"/>
    <xs:enumeration value="GRANT"/>
  </xs:restriction>
</xs:simpleType>

<!-- simple type for mandatory string
which must contain at least 1 character -->
<xs:simpleType name="mandatoryString">
  <xs:annotation>
    <xs:documentation>
      mandatoryType must contain at least 1 character
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="preserve"/>
    <xs:minLength value="1" />
  </xs:restriction>
</xs:simpleType>

<!-- simple type of a filesystem (file or folder) name -->
<xs:simpleType name="fsName">
  <xs:annotation>
    <xs:documentation>
      fsNames may only consist of ASCII characters and digits
      and must start with a non-digit
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="([a-z]|[A-Z])([a-z]|[A-Z]|[0-9]).*" />
    <xs:minLength value="1" />
  </xs:restriction>
</xs:simpleType>

```

```

<!-- simple type for action time of a trigger -->
<xs:simpleType name="actionTimeType">
  <xs:annotation>
    <xs:documentation>
      actionTime is BEFORE or AFTER
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="BEFORE" />
    <xs:enumeration value="AFTER" />
  </xs:restriction>
</xs:simpleType>

<!-- simple type for match type of a foreign key -->
<xs:simpleType name="matchTypeType">
  <xs:annotation>
    <xs:documentation>
      matchType is FULL, PARTIAL or SIMPLE
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="FULL" />
    <xs:enumeration value="PARTIAL" />
    <xs:enumeration value="SIMPLE" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

5.2 metadata.xml

Példa egy metadata.xml fájlra, amely megfelel a megadott XML séma-definíciónak:

```

<?xml version="1.0" encoding="UTF-8"?>
<siardArchive xmlns="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd
metadata.xsd">
  <dbname>jdbc:oracle:thin:@dbhost.enternet.ch:1521:SIARD1</dbname>
  <dataOwner>SIARD</dataOwner>
  <dataOriginTimespan>Fri May 16 11:21:39 CEST 2008</dataOriginTimespan>
  <archivalDate>2008-05-16</archivalDate>
  <messageDigest>MD5B9FB4FA23EFC27F10957533D747A4300</messageDigest>
  <clientMachine>blue2400.enterag.ch</clientMachine>
  <databaseProduct>
    Oracle Oracle9i Enterprise Edition Release 9.2.0.1.0 -
    Production\u0000AWith the Partitioning, OLAP and Oracle Data Mining
    options\u0000AJServer Release 9.2.0.1.0 - Production
  </databaseProduct>
  <connection>jdbc:oracle:thin:@dbhost.enternet.ch:1521:SIARD1</connection>
  <databaseUser>SIARD</databaseUser>
  <schemas>
    <schema>
      <name>SIARD</name>
      <folder>schema0</folder>
      <tables>
        <table>
          <name>TABLETEST</name>
          <folder>table0</folder>
          <description/>
          <columns>
            <column>
              <name>NID</name>
              <type>DECIMAL(38,0)</type>
              <typeOriginal>NUMBER</typeOriginal>
              <nullable>>false</nullable>
            </column>
            <column>
              <name>SNAME</name>
              <type>CHARACTER VARYING(31)</type>
            </column>
          </columns>
        </table>
      </tables>
    </schema>
  </schemas>

```



```

        <typeOriginal>VARCHAR2</typeOriginal>
        <nullable>>true</nullable>
    </column>
    <column>
        <name>TSCREATED</name>
        <type>DATE</type>
        <typeOriginal>DATE</typeOriginal>
        <nullable>>false</nullable>
    </column>
</columns>
<primaryKey>
    <name>TABLETEST_PK</name>
    <column>NID</column>
    <column>TSCREATED</column>
</primaryKey>
<candidateKeys>
    <candidateKey>
        <name>"Unique2"</name>
        <column>SNAME</column>
        <column>TSCREATED</column>
    </candidateKey>
    <candidateKey>
        <name>UNIQUE1</name>
        <column>TSCREATED</column>
    </candidateKey>
</candidateKeys>
<rows>2</rows>
</table>
<table>
    <name>"TableTest1"</name>
    <folder>table1</folder>
    <description/>
    <columns>
        <column>
            <name>"nID"</name>
            <type>DECIMAL(38,0)</type>
            <typeOriginal>NUMBER</typeOriginal>
            <nullable>>false</nullable>
        </column>
        <column>
            <name>"sName"</name>
            <type>CHARACTER VARYING(31)</type>
            <typeOriginal>VARCHAR2</typeOriginal>
            <nullable>>true</nullable>
        </column>
        <column>
            <name>"tsCreated"</name>
            <type>DATE</type>
            <typeOriginal>DATE</typeOriginal>
            <nullable>>false</nullable>
        </column>
    </columns>
    <primaryKey>
        <name>"TableTest1_PK"</name>
        <column>"nID"</column>
    </primaryKey>
    <foreignKeys>
        <foreignKey>
            <name>"TableTest1_FK"</name>
            <referencedSchema>SIARD</referencedSchema>
            <referencedTable>TABLETEST</referencedTable>
            <reference>
                <column>"nID"</column>
                <referenced>NID</referenced>
            </reference>
            <reference>
                <column>"tsCreated"</column>
                <referenced>TSCREATED</referenced>
            </reference>
            <deleteAction>RESTRICT</deleteAction>
            <updateAction>CASCADE</updateAction>
        </foreignKey>
    </foreignKeys>
    <candidateKeys>

```

```

        <candidateKey>
          <name>UNIQUE3</name>
          <column>"tsCreated"</column>
        </candidateKey>
      </candidateKeys>
      <rows>0</rows>
    </table>
    <table>
      <name>TABLETEST2</name>
      <folder>table2</folder>
      <description/>
      <columns>
        <column>
          <name>NID</name>
          <type>DECIMAL(38,0)</type>
          <typeOriginal>NUMBER</typeOriginal>
          <nullable>>false</nullable>
        </column>
        <column>
          <name>SNAME</name>
          <type>CHARACTER VARYING(31)</type>
          <typeOriginal>VARCHAR2</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>SCLOB</name>
          <folder>lob3</folder>
          <type>CHARACTER LARGE OBJECT(4000)</type>
          <typeOriginal>CLOB</typeOriginal>
          <nullable>>false</nullable>
        </column>
        <column>
          <name>TSCREATED</name>
          <type>DATE</type>
          <typeOriginal>DATE</typeOriginal>
          <nullable> false</nullable>
        </column>
        <column>
          <name>COLBINARYLARGEOBJECT</name>
          <folder>lob5</folder>
          <type>BINARY LARGE OBJECT(4000)</type>
          <typeOriginal>BLOB</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>COLCHARACTER</name>
          <type>CHARACTER(2)</type>
          <typeOriginal>CHAR</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>COLDATE</name>
          <type>DATE</type>
          <typeOriginal>DATE</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>COLDECIMAL</name>
          <type>DECIMAL(2,1)</type>
          <typeOriginal>NUMBER</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>COLDOUBLEPRECISION</name>
          <type>DECIMAL(22,0)</type>
          <typeOriginal>NUMBER</typeOriginal>
          <nullable>>true</nullable>
        </column>
        <column>
          <name>COLFLOAT</name>
          <type>DECIMAL(22,0)</type>
          <typeOriginal>NUMBER</typeOriginal>
          <nullable>>true</nullable>
        </column>
      </columns>
    </table>

```

```

    <column>
      <name>COLINTEGER</name>
      <type>DECIMAL(38,0)</type>
      <typeOriginal>NUMBER</typeOriginal>
      <nullable>>true</nullable>
    </column>
    <column>
      <name>COLNUMERIC</name>
      <type>DECIMAL(2,1)</type>
      <typeOriginal>NUMBER</typeOriginal>
      <nullable>true</nullable>
    </column>
    <column>
      <name>COLREAL</name>
      <type>DECIMAL(22,0)</type>
      <typeOriginal>NUMBER</typeOriginal>
      <nullable>true</nullable>
    </column>
    <column>
      <name>COLSMALLINT</name>
      <type>DECIMAL(38,0)</type>
      <typeOriginal>NUMBER</typeOriginal>
      <nullable>true</nullable>
    </column>
  </columns>
  <primaryKey>
    <name>TABLETEST2_PK</name>
    <column>NID</column>
  </primaryKey>
  <rows>4</rows>
</table>
</tables>
<views>
  <view>
    <name>"View1"</name>
    <columns>
      <column>
        <name>"nID"</name>
        <type>DECIMAL(38,0)</type>
        <typeOriginal>NUMBER</typeOriginal>
        <nullable>>false</nullable>
      </column>
      <column>
        <name>"sName"</name>
        <type>CHARACTER VARYING(31)</type>
        <typeOriginal>VARCHAR2</typeOriginal>
        <nullable>true</nullable>
      </column>
      <column>
        <name>"tsCreated"</name>
        <type>DATE</type>
        <typeOriginal>DATE</typeOriginal>
        <nullable>>false</nullable>
      </column>
    </columns>
  </view>
  <view>
    <name>VIEW2</name>
    <columns>
      <column>
        <name>"nID"</name>
        <type>DECIMAL(38,0)</type>
        <typeOriginal>NUMBER</typeOriginal>
        <nullable>>false</nullable>
      </column>
      <column>
        <name>"sName"</name>
        <type>CHARACTER VARYING(31)</type>
        <typeOriginal>VARCHAR2</typeOriginal>
        <nullable>true</nullable>
      </column>
      <column>
        <name>"tsCreated"</name>
        <type>DATE</type>

```

```

        <typeOriginal>DATE</typeOriginal>
        <nullable>>false</nullable>
    </column>
</columns>
</view>
</views>
<routines>
  <routine>
    <name>IS_LE</name>
    <description>
      Standalone procedure or function
    </description>
    <returnType>NUMERIC</returnType>
    <parameters>
      <parameter>
        <name>S1</name>
        <mode>IN</mode>
        <type>CHARACTER VARYING</type>
        <typeOriginal>VARCHAR2</typeOriginal>
      </parameter>
      <parameter>
        <name>S2</name>
        <mode>IN</mode>
        <type>CHARACTER VARYING</type>
        <typeOriginal>VARCHAR2</typeOriginal>
      </parameter>
    </parameters>
  </routine>
  <routine>
    <name>"IsZero"</name>
    <description>
      Standalone procedure or function
    </description>
    <returnType>NUMERIC</returnType>
    <parameters>
      <parameter>
        <name>I</name>
        <mode>IN</mode>
        <type>DECIMAL(38,0)</type>
        <typeOriginal>NUMBER</typeOriginal>
      </parameter>
    </parameters>
  </routine>
</routines>
</schema>
</schemas>
<users>
  <user>
    <name>SIARD3</name>
  </user>
  <user>
    <name>SIARD1</name>
  </user>
  <user>
    <name>SIARD2</name>
  </user>
  <user>
    <name>SIARD</name>
  </user>
</users>
<roles>
  <role>
    <name>"siardrole2"</name>
    <admin/>
  </role>
  <role>
    <name>SIARDROLE3</name>
    <admin/>
  </role>
  <role>
    <name>SIARDROLE1</name>
    <admin/>
  </role>
</roles>

```

```

    <name>RESOURCE</name>
    <admin/>
  </role>
</roles>
<privileges>
  <privilege>
    <type>SELECT</type>
    <object>TABLE TABLETEST2</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD2</grantee>
  </privilege>
  <privilege>
    <type>SELECT</type>
    <object>TABLE "TableTest1"</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD1</grantee>
  </privilege>
  <privilege>
    <type>SELECT</type>
    <object>TABLE "TableTest1"</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD3</grantee>
  </privilege>
  <privilege>
    <type>SELECT</type>
    <object>TABLE TABLETEST</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD2</grantee>
  </privilege>
  <privilege>
    <type>UPDATE</type>
    <object>TABLE TABLETEST2</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD2</grantee>
  </privilege>
  <privilege>
    <type>UPDATE</type>
    <object>TABLE TABLETEST2</object>
    <grantor>SIARD</grantor>
    <grantee>SIARD3</grantee>
  </privilege>
  <privilege>
    <type>SELECT</type>
    <object>TABLE "TableTest1"</object>
    <grantor>SIARD</grantor>
    <grantee>"siardrole2"</grantee>
  </privilege>
</privileges>
</siardArchive>

```

5.3 Példa egy tábla XML séma-definíciójára: table0.xsd

A SIARD minden táblára létrehoz egy XML séma definíciót, amely a megfelelő XML adat típust hozzárendeli az oszlopokhoz.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.admin.ch/xmlns/siard/1.0/schema0/table0.xsd" attributeFormDefault="unqualified" elementFormDefault="qualified" targetNamespace="http://www.admin.ch/xmlns/siard/1.0/schema0/table0.xsd">
  <xs:element name="table">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="row" type="rowType">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

</xs:element>
<xs:complexType name="rowType">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="c1" type="xs:decimal"/>
      <xs:element minOccurs="0" name="c2" type="xs:string"/>
      <xs:element name="c3" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

5.4 Példa egy tábla elsődleges adatára: table0.xml

Az elsődleges adat egy XML fájlban van, mely megfelel a tábla XML séma-definíciójának.

```

<?xml version="1.0" encoding="utf-8"?>
<table
  xsi:schemaLocation="http://www.admin.ch/xmlns/siard/1.0/schema0/table0.xsd
  table0.xsd"
  xmlns="http://www.admin.ch/xmlns/siard/1.0/schema0/table0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row><c1>1</c1><c2>First Name</c2><c3>2008-05-09</c3></row>
  <row><c1>2</c1><c2>Second Name</c2><c3>2008-05-10</c3></row>
</table>

```