

Specifiche del formato SIARD-2.2

Nome	Specifiche del formato SIARD-2.2
Categoria	Standard
Grado di maturità	Approvato
Versione	2.2
Stato	Versione rivista
Approvato il	2021-08-31
Pubblicato il	2021-08-31
Sostituisce	SIARD-2.1.1
Requisiti	Nessuno
Allegati	metadata.xsd
Lingue	Inglese, tedesco (traduzione), francese (traduzione), italiano (traduzione)
Editore / distributore	DILCIS Board, www.dilcis.eu Archivio federale svizzero, www.archivio-federale.ch

Sintesi

Il presente documento contiene le specifiche del formato di file SIARD, versione 2.2. L'acronimo SIARD sta per *Software-Independent Archival of Relational Databases* (archiviazione indipendente dal software di database relazionali). La versione 1.0 è stata elaborata dall'Archivio federale svizzero. Quella fornita nel seguito è la descrizione normativa di un formato di file per la conservazione a lungo termine di database relazionali.

Il formato SIARD si basa su standard, tra cui le norme ISO Unicode, XML e SQL:2008, lo standard Internet URI e lo standard industriale ZIP. L'applicazione di standard internazionalmente riconosciuti è tesa a garantire la conservazione a lungo termine e l'accesso al modello ampiamente diffuso di database relazionale e a permettere lo scambio semplice di contenuti di database a prescindere dai formati *dump* proprietari.

Cronologia delle versioni

Rapporto tra la presente versione e le versioni precedenti:

- eCH-0165 v1.0 sostituita dalla versione 2.1.1
La versione 1.0 è stata sostituita da una versione più recente conforme allo standard vigente. È ancora possibile usarla, ma si raccomanda di impiegare la presente versione.
- eCH-0165 v2.0 soppressa
La versione 2.0 è stata sostituita a causa di errori e ambiguità che avrebbero potuto causare problemi pratici sul lungo periodo. La versione 2.0 non può più essere usata. Devono essere impiegate la presente versione o, in alternativa, la versione 1.0.
- SIARD-2.1 sostituita dalla versione 2.1.1
Nella versione 2.1 sono stati eliminati errori e ambiguità della versione 2.0. È stata sviluppata dal gruppo Archiviazione digitale di eCH, ma non è uno standard ufficiale eCH.
- SIARD-2.1.1 sostituita dalla versione 2.2
La versione 2.1.1 costituisce lo stato attuale del formato SIARD. Fatte salve alcune poche precisazioni nella formulazione, è identica alla versione 2.1.
- SIARD-2.2 La versione 2.2 supporta anche la gestione di file memorizzati al di fuori del database in conformità alla parte 9 di SQL:2008 (ISO/IEC 9075-9:2008 – SQL/MED) e la scalabilità per gestire grandi oggetti memorizzati al di fuori del file SIARD. Per il resto è identica alla versione 2.1.1.
È stata sviluppata dal DILCIS Board nel quadro del progetto E-ARK3¹.

¹ SIARD 2.2 dovrebbe essere supportato da SIARD Suite, Database Preservation Toolkit e altre applicazioni.

Indice

1	Introduzione.....	5
1.1	Stato	5
1.2	Campo d'applicazione	5
2	Struttura del presente documento	7
2.1	Capitoli	7
2.2	ID dei requisiti.....	7
2.3	Distinzione tra requisiti obbligatori e facoltativi.....	8
2.4	Notazione di cartelle, file e strutture di cartelle.....	8
3	Principi / requisiti generali.....	9
3.1	Applicazione di standard.....	9
3.2	Database sotto forma di documenti	9
3.3	Set di caratteri e caratteri.....	9
3.4	Schema dell'URI di file.....	10
3.5	Identificatori e identificatori regolari.....	11
4	Requisiti per la struttura del formato	12
4.1	Struttura del file d'archivio SIARD	12
4.2	Struttura del file d'archivio SIARD.....	12
4.3	Corrispondenza tra metadati e dati tabellari.....	15
5	Requisiti per i metadati	20
5.1	Metadati a livello di database.....	20
5.2	Metadati a livello di schema.....	22
5.3	Metadati a livello di tipo.....	22
5.4	Metadati a livello di attributo	24
5.5	Metadati a livello di tabella.....	25
5.6	Metadati a livello di colonna.....	26
5.7	Metadati dei campi.....	28
5.8	Metadati della chiave primaria	29
5.9	Metadati delle chiavi esterne.....	29
5.10	Metadati delle referenze	30
5.11	Metadati della chiave candidata.....	30
5.12	Metadati del vincolo di controllo	30
5.13	Metadati a livello di trigger	31
5.14	Metadati a livello di vista	31
5.15	Metadati a livello di routine	32
5.16	Metadati dei parametri	33
5.17	Metadati a livello di utente	34
5.18	Metadati a livello di ruolo	35
5.19	Metadati a livello di privilegi	35
6	Requisiti per i dati tabellari.....	36
6.1	Definizione dello schema delle tabelle	36
6.2	Celle di dati <i>large object</i>	37

6.3	Celle di dati <i>data</i> e <i>timestamp</i>	38
6.4	Dati tabellari.....	38
7	Requisiti per la struttura delle cartelle per LOB memorizzati al di fuori del file SIARD	40
7.1	Struttura delle cartelle per LOB memorizzati al di fuori del file SIARD.....	40
8	Problemi di scalabilità	42
8.1	Segmentazione di LOB memorizzati al di fuori del file SIARD.....	42
8.2	Suddivisione del file SIARD in parti binarie	45
9	Versione e validità delle specifiche	46
10	Processo di gestione delle modifiche	46
11	Esclusione della responsabilità e diritti di terzi	46
12	Diritti d'autore	46
	Allegato A – Collaborazione e verifica	47
	Allegato B – Abbreviazioni e glossario	48
	Allegato C – Documentazione degli standard applicati	50
	Allegato D – Estratti dell'esempio ech-0165_oe.siard	51
	Allegato E – Esempio di segmentazione di LOB interni	79
	Allegato F – Gestione di <i>large object</i> in database relazionali	83
	Allegato G – SIARD e ZIP	85
	Allegato H – Modifiche rispetto alla versione 1.0	86

1 Introduzione

1.1 Stato

Il presente documento è stato approvato dal DILCIS Board e dall'Archivio federale svizzero.

1.2 Campo d'applicazione

1.2.1 Destinatari

Il presente è un documento tecnico destinato a specialisti IT che lavorano nel campo dell'archiviazione permanente di database relazionali.

1.2.2 Contesto

SIARD è l'acronimo di *Software-Independent Archival of Relational Databases* (archiviazione indipendente dal software di database relazionali), un formato di file aperto per l'archiviazione permanente di database relazionali sotto forma di dati testuali basati su XML, impacchettati in un file contenitore (archivio SIARD)².

Per archiviazione permanente s'intende la conservazione a tempo indeterminato delle informazioni memorizzate nei file SIARD preservando la sequenza di bit e garantendo l'interpretabilità e la presentazione dei dati in una forma leggibile e comprensibile per l'essere umano.

Grazie alla traduzione della struttura e del contenuto di un database relazionale nel formato SIARD sarà possibile accedere in ogni momento ai dati memorizzati o scambiarli, anche se il software originale del database non sarà più disponibile od operativo. A tal fine si applicano standard appropriati per il formato SIARD che godono di un ampio sostegno internazionale. Questa interpretabilità a lungo termine dei contenuti dei database poggia essenzialmente sui due standard XML e SQL:2008.

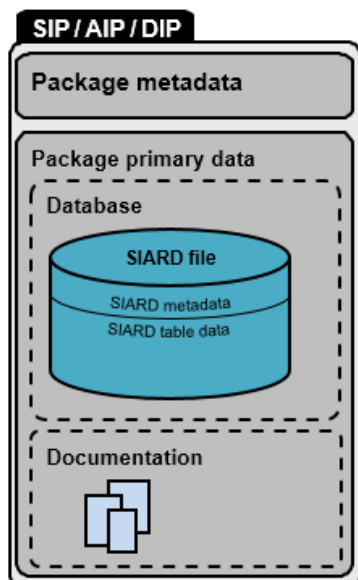
² Il formato SIARD per l'archiviazione di database va distinto dall'applicazione SIARD Suite sviluppata dall'Archivio federale svizzero per generare, modificare e reimportare file SIARD in ambienti database.

1.2.3 Restrizioni

Il formato SIARD si presta unicamente all'archiviazione a lungo termine di un tipo specifico di documenti digitali (database relazionali) ed è quindi concepito in modo del tutto indipendente dalle strutture dei pacchetti, quali SIP (*Submission Information Package*), AIP (*Archival Information Package*) e DIP (*Dissemination Information Package*), del modello OAIS³.

Il presupposto è che un database in formato SIARD sia archiviato come parte di un tale pacchetto informativo insieme ad altri documenti (*large object file* memorizzati esternamente, tabella per la traduzione dei nomi di file esterni, documentazione sul database, documenti aziendali rilevanti per la comprensione del database ecc.).

Analogamente a un file di posta elettronica o Word basato su XML, che presenta una struttura interna con metadati, dati primari e diversi dati ausiliari, un database relazionale archiviato in formato SIARD contiene, oltre ai dati tabellari, metadati propri che descrivono il documento in modo dettagliato e che sono indipendenti dal catalogo di metadati che un archivio include nei suoi pacchetti OAIS.



³ <http://public.ccsds.org/pubs/650x0m2.pdf>

2 Struttura del presente documento

2.1 Capitoli

Tutti i capitoli presentano la stessa struttura. Dopo una breve introduzione, sono elencati in forma tabellare i requisiti.

ID	Descrizione requisito	O/F
ID del requisito	Testo del requisito	Carattere obbligatorio (O) o facoltativo (F) del requisito

La descrizione di un requisito è spesso corredata di raccomandazioni, note ed esempi, designati come tali.

ID	Descrizione requisito	O/F
G_2.1-1	Testo del requisito Esempio Testo dell'esempio Nota Testo della nota Raccomandazione <i>Il testo della raccomandazione è in corsivo.</i>	O

2.2 ID dei requisiti

I requisiti sono identificabili in modo univoco con un ID.

ID
G_2.2-1

L'ID è strutturato nel modo seguente:

G_	lettera + _	identifica il capitolo principale
G_	=	principi / requisiti generali
T_	=	requisiti per i dati tabellari
M_	=	requisiti per i metadati
P_	=	requisiti per la struttura del pacchetto
L_	=	requisiti per la struttura di LOB memorizzati al di fuori del file SIARD
S_	=	requisiti per la segmentazione di LOB e del file SIARD
2.2-1	Il primo numero corrisponde al numero del capitolo. Ogni capitolo è dedicato a una categoria di requisiti. Il numero dopo il trattino è progressivo e identifica i singoli requisiti del capitolo.	

2.3 Distinzione tra requisiti obbligatori e facoltativi

I requisiti si suddividono in due categorie: quelli obbligatori e quelli facoltativi. Il carattere obbligatorio o facoltativo è contrassegnato dalle seguenti lettere:

Lettera	Significato
O	Requisito obbligatorio Il requisito deve essere soddisfatto per ottenere un file SIARD valido.
F	Requisito facoltativo Il requisito dovrebbe essere soddisfatto. Semplifica la gestione e costituisce una buona pratica.

2.4 Notazione di cartelle, file e strutture di cartelle

Per la notazione di cartelle, file ecc. sono usati i seguenti simboli e parametri:

Simbolo	Significato
/	Cartella
header/	Cartella denominata <i>header</i>
xy.txt	File (con l'estensione .txt)
dir1/	Cartella di esempio (in rosso)
abc.pdf	File di esempio (in rosso)
...	Segnaposto per file o cartelle non rilevanti per la spiegazione
[]	Segnaposto per un'espressione o un tipo base come <i>string</i> , <i>integer</i> ecc.
<xx>	Segnaposto per qualsiasi stringa di caratteri

3 Principi / requisiti generali

3.1 Applicazione di standard

Per garantire l'interpretabilità sul lungo termine dei contenuti dei database, il formato SIARD poggia essenzialmente sui due standard ISO XML e SQL:2008.

ID	Descrizione requisito	O/F
G_3.1-1	Tutti i contenuti del database sono memorizzati in una collezione di file in formato XML 1.0 ⁴ secondo le definizioni dello schema XML 1.0 ⁵ . Le definizioni e il codice SQL devono essere conformi a SQL:2008 secondo ISO/IEC 9075. Le uniche eccezioni sono i dati BLOB (<i>Binary Large Object</i>) e CLOB (<i>Character Large Object</i>), memorizzati in file binari o di testo separati, ma referenziati nei file XML.	O

3.2 Database sotto forma di documenti

Un database relazionale è trattato come un unico documento da archiviare in modo che siano preservate le relazioni (le referenze) tra i dati delle diverse tabelle.

ID	Descrizione requisito	O/F
G_3.2-1	Un database relazionale è archiviato in un unico file SIARD in cui possono essere eventualmente referenziati <i>large object</i> memorizzati esternamente, ma che in senso lato fanno parte del database. In rari casi potrebbe essere necessario segmentare il file SIARD a causa delle sue dimensioni.	O

3.3 Set di caratteri e caratteri

ID	Descrizione requisito	O/F
G_3.3-1	Tutti i dati sono memorizzati usando il set di caratteri Unicode, in conformità a ISO 10646.	O
G_3.3-2	I dati estratti da database che supportano altri set di caratteri sono rappresentati nei set di caratteri Unicode corrispondenti. Per questo motivo, i tipi di stringhe di caratteri nazionali estratti (NCHAR, NCHAR VARYING, NCLOB) devono essere generalmente tradotti in tipi non nazionali (CHAR, VARCHAR o CLOB). Questa convenzione è supportata da XML sia per i file XML memorizzati nel formato UTF-8 sia per quelli memorizzati nel formato UTF-16.	O
G_3.3-3	Nei file XML del formato SIARD, tutti i caratteri con un significato speciale nella sintassi XML sono sostituiti da entità carattere in tutti i campi di tipo xs:string. I caratteri di controllo Unicode 0-31 e 127-159 sono inoltre codificati utilizzando la barra inversa (\) per garantire la validità del file XML.	O

⁴ <https://www.w3.org/TR/REC-xml/>

⁵ <https://www.w3.org/TR/xmlschema-1/>, <https://www.w3.org/TR/xmlschema-2/>,
<https://www.w3.org/TR/xmlschema-ref/>

ID	Descrizione requisito	O/F																						
G_3.3-4	<p>I caratteri che non possono essere rappresentati in Unicode (codici 0-8, 14-31, 127-159), come pure il carattere di escape (\) e diversi spazi vuoti consecutivi sono rappresentati in XML mediante la combinazione \u00<xx>. Le virgolette e i segni <, > e & sono rappresentati in XML come entità carattere.</p> <table border="1"> <thead> <tr> <th>Caratteri originali</th> <th>Caratteri nel formato SIARD</th> </tr> </thead> <tbody> <tr> <td>0-8</td> <td>\u0000-\u0008</td> </tr> <tr> <td>14-31</td> <td>\u000E-\u001F</td> </tr> <tr> <td>32</td> <td>\u0020, se diversi spazi vuoti consecutivi</td> </tr> <tr> <td>"</td> <td>&quot;</td> </tr> <tr> <td>&</td> <td>&amp;</td> </tr> <tr> <td>'</td> <td>&apos;</td> </tr> <tr> <td><</td> <td>&lt;</td> </tr> <tr> <td>></td> <td>&gt;</td> </tr> <tr> <td>\</td> <td>\u005c</td> </tr> <tr> <td>127-159</td> <td>\u007F-\u009F</td> </tr> </tbody> </table>	Caratteri originali	Caratteri nel formato SIARD	0-8	\u0000-\u0008	14-31	\u000E-\u001F	32	\u0020, se diversi spazi vuoti consecutivi	"	"	&	&	'	'	<	<	>	>	\	\u005c	127-159	\u007F-\u009F	O
Caratteri originali	Caratteri nel formato SIARD																							
0-8	\u0000-\u0008																							
14-31	\u000E-\u001F																							
32	\u0020, se diversi spazi vuoti consecutivi																							
"	"																							
&	&																							
'	'																							
<	<																							
>	>																							
\	\u005c																							
127-159	\u007F-\u009F																							

3.4 Schema dell'URI di file

Per referenziare *large object* memorizzati esternamente è usato lo schema dell'URI di file, in conformità a RFC 8089⁶.

ID	Descrizione requisito	O/F
G_3.4-1	Tutti i file memorizzati esternamente sono specificati usando un URI di file, in conformità a RFC 8089.	O
G_3.4-2	Gli URI di file sono memorizzati in un file SIARD sotto forma di stringhe ASCII con codifica URL.	O
G_3.4-3	Se l'URI di file si basa su un sistema di file che permette l'indirizzamento diretto dei singoli file all'interno di un file ZIP, i <i>large object</i> memorizzati esternamente possono essere integrati in file ZIP. Per esempio, <i>file:///d:/sips/sips/sip1234.zip</i> rimanda al file ZIP, mentre <i>file:///d:/sips/sip1234.zip/</i> alla cartella radice all'interno del file ZIP.	F

⁶ http://en.wikipedia.org/wiki/File_URI_scheme, <http://tools.ietf.org/html/rfc8089>

3.5 Identificatori e identificatori regolari

In SQL:2008 ci sono identificatori regolari⁷ senza spazi né caratteri speciali, per i quali le lettere maiuscole e minuscole non giocano alcun ruolo, ma che sono memorizzati nel file SIARD in lettere maiuscole, e identificatori tra virgolette⁸, per i quali la scrittura maiuscola e minuscola è univoca e che possono anche contenere caratteri speciali o essere identici a una parola chiave SQL. Questi ultimi devono essere posti tra virgolette doppie. Nel file SIARD sono memorizzati senza le virgolette.

Lo standard SQL definisce che cosa è un carattere speciale o una parola chiave. Lo standard Unicode definisce invece qual è la versione maiuscola di una lettera.

Nei metadati, l'identificatore regolare è memorizzato in lettere maiuscole, mentre tutti gli altri identificatori sono memorizzati invariati senza virgolette. Secondo lo standard SQL:2008, un identificatore deve essere considerato delimitato se contiene un carattere che in quanto identificatore regolare non può contenere o se è identico a una parola chiave SQL.

ID	Descrizione requisito	O/F
G_3.5-1	Tutti gli identificatori sono memorizzati usando il set di caratteri Unicode.	O
G_3.5-2	Gli identificatori regolari sono in lettere maiuscole e senza virgolette.	O
G_3.5-3	Gli identificatori delimitati sono memorizzati senza virgolette.	O

⁷ Identificatore regolare, ingl.: *identifier*. Un identificatore SQL:2008 deve iniziare con una lettera (A-Z) o un trattino basso (), seguito da lettere (A-Z), cifre (0-9) o un trattino basso () e avere una lunghezza massima di 128 caratteri.

⁸ Identificatore tra virgolette o identificatore delimitato, ingl.: *delimited identifier*.

4 Requisiti per la struttura del formato

4.1 Struttura del file d'archivio SIARD

Il file d'archivio SIARD è strutturato come archivio ZIP.

ID	Descrizione requisito	O/F
G_4.1-1	Il file SIARD è memorizzato come singolo archivio ZIP, in conformità alle specifiche pubblicate dalla ditta PkWare, versione 6.3.2 ⁹ .	O
G_4.1-2	I file SIARD non devono essere compressi oppure essere compressi con l'algoritmo deflate, in conformità a RFC 1951 ¹⁰ . Raccomandazione <i>Si raccomanda di usare l'algoritmo deflate.</i>	O
G_4.1-3	Il file SIARD non è né protetto da password né crittografato.	O
G_4.1-4	Per l'archivio ZIP è ammesso sia ZIP32 sia ZIP64.	O
G_4.1-5	L'archivio ZIP ha l'estensione <code>.siard</code> .	O

4.2 Struttura del file d'archivio SIARD

Un database relazionale archiviato nel formato SIARD è costituito da due componenti: i metadati, che ne descrivono la struttura, e i dati tabellari, che rappresentano i contenuti delle tabelle. I metadati indicano inoltre quali dati tabellari si trovano dove nell'archivio.

ID	Descrizione requisito	O/F
P_4.2-1	I dati tabellari si trovano nella cartella <code>content/</code> e i metadati nella cartella <code>header/</code> . Non sono ammessi altri file o altre cartelle nella cartella radice. Esempio Struttura (schematica) del file SIARD <pre>ech-0165_oe.siard content/ header/</pre>	O

⁹ Definiti originariamente da Phil Katz, i file ZIP sono di fatto uno standard oggi molto diffuso. L'attuale versione 6.3.9 delle specifiche pubblicate dalla ditta PkWare è disponibile all'indirizzo <https://support.pkware.com/display/PKZIP/Application+Note+Archives>.

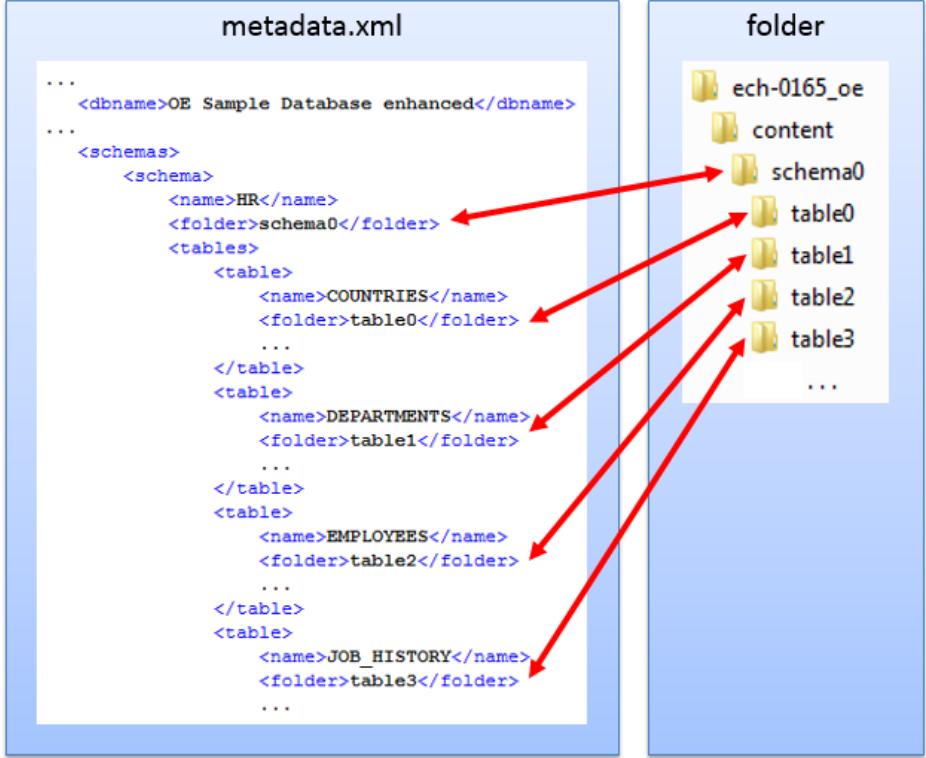

¹⁰ <https://www.ietf.org/rfc/rfc1951.txt>.

ID	Descrizione requisito	O/F
P_4.2-2	<p>La cartella <code>content/</code> contiene una o più cartelle di schema in cui si trovano le cartelle delle singole tabelle. Non sono ammessi altri file o altre cartelle.</p> <p>Esempio Struttura (schematica) del file SIARD</p> <pre data-bbox="432 479 1318 786"> ech-0165_oe.siard content/ schema0/ table0/ table1/ table2/ ... schema1/ table0/ ... </pre> <p>Raccomandazione <i>Si raccomanda di uniformare i nomi delle cartelle degli schemi e delle tabelle e di usare, al posto del nome originale, per esempio <code>schema0/</code> e <code>table0/</code> (v. restrizioni in P_4.2-6).</i></p>	O
P_4.2-3	<p>Le cartelle delle singole tabelle contengono un file XML e un file XSD: i nomi (della cartella e dei due file) devono essere identici. Non sono ammessi altri file o altre cartelle, eccezion fatta per le cartelle BLOB e CLOB e i loro contenuti (file BIN, TXT, XML o, se il tipo MIME dei file LOB è noto, l'estensione associata a quest'ultimo, p. es. JPG).</p> <p>Esempio Struttura (schematica) del file SIARD</p> <pre data-bbox="432 1234 1318 1720"> ech-0165_oe.siard content/ ... schema1/ ... table6/ table6.xml table6.xsd table7/ table7.xml table7.xsd lob1¹¹/ record0.xml record1.xml ... </pre> <p>Raccomandazione <i>Si raccomanda di uniformare i nomi delle cartelle e dei file LOB e di usare, al posto del nome originale, per esempio <code>lob1/</code> e <code>record0.bin</code>, <code>record0.txt</code>, <code>record0.xml</code> o, se è conosciuto il tipo MIME dei file, l'estensione associata a quest'ultimo (v. restrizioni in P_4.2-6).</i></p>	O

¹¹ In questo esempio, la colonna 2 contiene file LOB supplementari memorizzati in `lob1/`.

ID	Descrizione requisito	O/F
P_4.2-4	Per facilitare il riconoscimento del formato SIARD (p. es. mediante PRONOM), il file deve contenere una cartella vuota <code>header/siardversion/2.2/</code> che indica la versione del formato SIARD.	O
P_4.2-5	<p>La cartella <code>header/</code> deve contenere i file <code>metadata.xml</code> e <code>metadata.xsd</code>. Sono ammessi altri file, per esempio fogli di stile.</p> <p>Esempio Struttura (schematica) del file SIARD</p> <pre data-bbox="432 600 1318 882"> ech-0165_oe.siard content/ ... header/ metadata.xml metadata.xsd siardversion/ 2.2/ ... </pre>	O
P_4.2-6	<p>I nomi dei file e delle cartelle devono essere strutturati come segue: devono iniziare con una lettera (a-z o A-Z) e possono contenere soltanto i seguenti caratteri:</p> <ul style="list-style-type: none"> • a-z • A-Z • 0-9 • - • . (può essere usato soltanto per separare il nome dall'estensione) <p>Raccomandazione <i>Nel limite del possibile, la lunghezza dei nomi dei file e delle cartelle non deve superare i 20 caratteri per evitare problemi con percorsi troppo lunghi in Windows.</i></p>	O

4.3 Corrispondenza tra metadati e dati tabellari


ID	Descrizione requisito	O/F
P_4.3-1	<p>La struttura definita in metadata.xml deve essere identica a quella della cartella content/.</p> <p>Esempio</p>  <pre> ... <dbname>OE Sample Database enhanced</dbname> ... <schemas> <schema> <name>HR</name> <folder>schema0</folder> </schema> <tables> <table> <name>COUNTRIES</name> <folder>table0</folder> </table> ... <table> <name>DEPARTMENTS</name> <folder>table1</folder> </table> ... <table> <name>EMPLOYEES</name> <folder>table2</folder> </table> ... <table> <name>JOB_HISTORY</name> <folder>table3</folder> </table> </tables> </schemas> ... </pre>	O
P_4.3-2	<p>Il numero di colonne di una tabella indicato in metadata.xml deve essere identico a quello della tabella corrispondente table[numero].xsd.</p> <p>Esempio</p>  <pre> ... <table> <name>COUNTRIES</name> <folder>table0</folder> <description> <columns> <column> <column> <column> </columns> <primaryKey> <rows>25</rows> </table> ... </pre> <pre> <?xml version="1.0" encoding="utf-8" standalone="no" ?> <xs:schema xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.x" <xs:element name="table"> ... <xs:complexType name="recordType"> ... <xs:sequence> <xs:element name="c1" type="xs:string"/> <xs:element minOccurs="0" name="c2" type="xs:string"/> <xs:element minOccurs="0" name="c3" type="xs:decimal"/> </xs:sequence> </xs:complexType> ... </xs:schema> </pre>	O

ID	Descrizione requisito	O/F																																																						
P_4.3-3	<p>Nelle definizioni delle colonne in <code>metadata.xml</code>, le informazioni sul tipo di dati devono essere identiche a quelle del file corrispondente <code>table[numero].xsd</code>.</p> <p>Nei file degli schemi <code>table[numero].xsd</code>, i tipi predefiniti di dati SQL:2008¹² sono convertiti in tipi di dati XML in base alla tabella seguente:</p> <table border="1" data-bbox="440 472 1370 1644"> <thead> <tr> <th data-bbox="440 472 1027 521">SQL:2008</th> <th data-bbox="1027 472 1370 521">XML</th> </tr> </thead> <tbody> <tr><td data-bbox="440 521 1027 555">BIGINT</td><td data-bbox="1027 521 1370 555">xs:integer</td></tr> <tr><td data-bbox="440 555 1027 589">BINARY LARGE OBJECT(...), BLOB(...)</td><td data-bbox="1027 555 1370 589">blobType¹³</td></tr> <tr><td data-bbox="440 589 1027 622">BINARY VARYING(...), VARBINARY(...)</td><td data-bbox="1027 589 1370 622">xs:hexBinary / clobType¹³</td></tr> <tr><td data-bbox="440 622 1027 656">BINARY(...)</td><td data-bbox="1027 622 1370 656">xs:hexBinary / blobType¹³</td></tr> <tr><td data-bbox="440 656 1027 689">BOOLEAN</td><td data-bbox="1027 656 1370 689">xs:boolean</td></tr> <tr><td data-bbox="440 689 1027 723">CHARACTER LARGE OBJECT(...), CLOB(...)</td><td data-bbox="1027 689 1370 723">clobType¹³</td></tr> <tr><td data-bbox="440 723 1027 757">CHARACTER VARYING(...), CHAR VARYING(...), VARCHAR(...)</td><td data-bbox="1027 723 1370 757">xs:string / clobType¹³</td></tr> <tr><td data-bbox="440 757 1027 790">CHARACTER(...), CHAR(...)</td><td data-bbox="1027 757 1370 790">xs:string / clobType¹³</td></tr> <tr><td data-bbox="440 790 1027 824">DATE</td><td data-bbox="1027 790 1370 824">dateTime</td></tr> <tr><td data-bbox="440 824 1027 857">DATALINK</td><td data-bbox="1027 824 1370 857">blobType / clobType¹³</td></tr> <tr><td data-bbox="440 857 1027 891">DECIMAL(...), DEC(...)</td><td data-bbox="1027 857 1370 891">xs:decimal</td></tr> <tr><td data-bbox="440 891 1027 925">DOUBLE PRECISION</td><td data-bbox="1027 891 1370 925">xs:double</td></tr> <tr><td data-bbox="440 925 1027 958">FLOAT(p)</td><td data-bbox="1027 925 1370 958">xs:double</td></tr> <tr><td data-bbox="440 958 1027 992">INTEGER, INT</td><td data-bbox="1027 958 1370 992">xs:integer</td></tr> <tr><td data-bbox="440 992 1027 1025">INTERVAL <start> [TO <end>]</td><td data-bbox="1027 992 1370 1025">xs:duration</td></tr> <tr><td data-bbox="440 1025 1027 1059">NATIONAL CHARACTER LARGE OBJECT(...), NCHAR LARGE OBJECT(...), NCLOB(...)</td><td data-bbox="1027 1025 1370 1059">clobType¹³</td></tr> <tr><td data-bbox="440 1059 1027 1093">NATIONAL CHARACTER VARYING(...), NATIONAL CHAR VARYING(...), NCHAR VARYING(...)</td><td data-bbox="1027 1059 1370 1093">xs:string / clobType¹³</td></tr> <tr><td data-bbox="440 1093 1027 1126">NATIONAL CHARACTER(...), NCHAR(...), NATIONAL CHAR(...),</td><td data-bbox="1027 1093 1370 1126">xs:string / clobType¹³</td></tr> <tr><td data-bbox="440 1126 1027 1160">NUMERIC(...)</td><td data-bbox="1027 1126 1370 1160">xs:decimal</td></tr> <tr><td data-bbox="440 1160 1027 1193">REAL</td><td data-bbox="1027 1160 1370 1193">xs:float</td></tr> <tr><td data-bbox="440 1193 1027 1227">SMALLINT</td><td data-bbox="1027 1193 1370 1227">xs:integer</td></tr> <tr><td data-bbox="440 1227 1027 1261">TIME(...)</td><td data-bbox="1027 1227 1370 1261">timeType</td></tr> <tr><td data-bbox="440 1261 1027 1294">TIME WITH TIME ZONE(...)</td><td data-bbox="1027 1261 1370 1294">timeType</td></tr> <tr><td data-bbox="440 1294 1027 1328">TIMESTAMP(...)</td><td data-bbox="1027 1294 1370 1328">dateTimeType</td></tr> <tr><td data-bbox="440 1328 1027 1361">TIMESTAMP WITH TIME ZONE(...)</td><td data-bbox="1027 1328 1370 1361">dateTimeType</td></tr> <tr><td data-bbox="440 1361 1027 1395">XML</td><td data-bbox="1027 1361 1370 1395">clobType¹³</td></tr> </tbody> </table>	SQL:2008	XML	BIGINT	xs:integer	BINARY LARGE OBJECT(...), BLOB(...)	blobType ¹³	BINARY VARYING(...), VARBINARY(...)	xs:hexBinary / clobType ¹³	BINARY(...)	xs:hexBinary / blobType ¹³	BOOLEAN	xs:boolean	CHARACTER LARGE OBJECT(...), CLOB(...)	clobType ¹³	CHARACTER VARYING(...), CHAR VARYING(...), VARCHAR(...)	xs:string / clobType ¹³	CHARACTER(...), CHAR(...)	xs:string / clobType ¹³	DATE	dateTime	DATALINK	blobType / clobType ¹³	DECIMAL(...), DEC(...)	xs:decimal	DOUBLE PRECISION	xs:double	FLOAT(p)	xs:double	INTEGER, INT	xs:integer	INTERVAL <start> [TO <end>]	xs:duration	NATIONAL CHARACTER LARGE OBJECT(...), NCHAR LARGE OBJECT(...), NCLOB(...)	clobType ¹³	NATIONAL CHARACTER VARYING(...), NATIONAL CHAR VARYING(...), NCHAR VARYING(...)	xs:string / clobType ¹³	NATIONAL CHARACTER(...), NCHAR(...), NATIONAL CHAR(...),	xs:string / clobType ¹³	NUMERIC(...)	xs:decimal	REAL	xs:float	SMALLINT	xs:integer	TIME(...)	timeType	TIME WITH TIME ZONE(...)	timeType	TIMESTAMP(...)	dateTimeType	TIMESTAMP WITH TIME ZONE(...)	dateTimeType	XML	clobType ¹³	O
SQL:2008	XML																																																							
BIGINT	xs:integer																																																							
BINARY LARGE OBJECT(...), BLOB(...)	blobType ¹³																																																							
BINARY VARYING(...), VARBINARY(...)	xs:hexBinary / clobType ¹³																																																							
BINARY(...)	xs:hexBinary / blobType ¹³																																																							
BOOLEAN	xs:boolean																																																							
CHARACTER LARGE OBJECT(...), CLOB(...)	clobType ¹³																																																							
CHARACTER VARYING(...), CHAR VARYING(...), VARCHAR(...)	xs:string / clobType ¹³																																																							
CHARACTER(...), CHAR(...)	xs:string / clobType ¹³																																																							
DATE	dateTime																																																							
DATALINK	blobType / clobType ¹³																																																							
DECIMAL(...), DEC(...)	xs:decimal																																																							
DOUBLE PRECISION	xs:double																																																							
FLOAT(p)	xs:double																																																							
INTEGER, INT	xs:integer																																																							
INTERVAL <start> [TO <end>]	xs:duration																																																							
NATIONAL CHARACTER LARGE OBJECT(...), NCHAR LARGE OBJECT(...), NCLOB(...)	clobType ¹³																																																							
NATIONAL CHARACTER VARYING(...), NATIONAL CHAR VARYING(...), NCHAR VARYING(...)	xs:string / clobType ¹³																																																							
NATIONAL CHARACTER(...), NCHAR(...), NATIONAL CHAR(...),	xs:string / clobType ¹³																																																							
NUMERIC(...)	xs:decimal																																																							
REAL	xs:float																																																							
SMALLINT	xs:integer																																																							
TIME(...)	timeType																																																							
TIME WITH TIME ZONE(...)	timeType																																																							
TIMESTAMP(...)	dateTimeType																																																							
TIMESTAMP WITH TIME ZONE(...)	dateTimeType																																																							
XML	clobType ¹³																																																							

¹² I tipi di dati BIT e BIT VARYING corrispondono a vecchie definizioni SQL sostituite con BOOLEAN e BINARY in SQL:2008. BIT(1) è convertito in BOOLEAN e BIT(n) in BINARY((n+7)/8).

¹³ Per i tipi di dati XML *blobType* e *clobType* si veda G_3.1-1, T_6.2-1 e `metadata.xsd`. DATALINK è rappresentato in XML come *blobType* (o *clobType*) usando l'attributo facoltativo DLURLPATHONLY.

ID	Descrizione requisito	O/F
	<p>Esempio</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="432 331 823 734" style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">metadata.xml</p> <pre style="font-family: monospace; font-size: 0.8em; margin: 0;"> ... <table> <name>COUNTRIES</name> <folder>table0</folder> <description> <columns> <column> <name>COUNTRY_ID</name> <type>CHARACTER(2)</type> <typeOriginal>"CHAR"</typeOriginal> <nullable>false</nullable> <description> </column> ... <column> <name>REGION_ID</name> <type>DECIMAL(22)</type> <typeOriginal>"NUMBER"</typeOriginal> <description> </column> ... </pre> </div> <div data-bbox="839 331 1366 734" style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">table0.xsd</p> <pre style="font-family: monospace; font-size: 0.8em; margin: 0;"> <?xml version="1.0" encoding="utf-8" standalone="no" ?> <xs:schema xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.x" <xs:element name="table"> ... <xs:complexType name="recordType"> <xs:sequence> <xs:element name="c1" type="xs:string"/> <xs:element minOccurs="0" name="c2" type="xs:string"/> <xs:element minOccurs="0" name="c3" type="xs:decimal"/> </xs:sequence> </xs:complexType> ... </xs:schema> </pre> </div> </div> <p>Tutti i tipi DATE e TIME sono specificati nel fuso orario UTC. Si raccomanda di farli terminare con una Z.</p> <p>dateType è una restrizione di xs:date in UTC agli anni compresi tra 0001 e 9999 (restrizione SQL:2008).</p> <p>timeType è una restrizione di xs:date al fuso orario UTC.</p> <p>dateTimeType è una restrizione di xs:dateTime nel fuso orario UTC agli anni compresi tra 0001 e 9999 (restrizione SQL:2008).</p> <p>clobType è un'estensione di xs:string. I valori inline non necessitano di attributi supplementari e il valore CLOB è memorizzato direttamente.</p> <p>Se i valori CLOB sono memorizzati in un file separato, è obbligatorio impostare gli attributi <i>file</i> e <i>length</i>. Oltre a questi attributi obbligatori, ci sono gli attributi facoltativi <i>digestType</i> e <i>digest</i>. Il valore dell'attributo <i>file</i> contiene l'URI del file (con codifica URL e se possibile riferito alla lobFolder più vicina) in cui è memorizzato il CLOB. Il valore dell'attributo <i>length</i> indica la lunghezza in UTF-8 e l'attributo facoltativo <i>digest</i> contiene l'informazione di integrità secondo l'attributo facoltativo <i>digestType</i>.</p> <p>blobType è un'estensione di xs:hexBinary. I valori inline non necessitano di attributi supplementari e il valore BLOB è memorizzato direttamente.</p> <p>Se i valori BLOB sono memorizzati in un file separato, è obbligatorio impostare gli attributi <i>file</i> e <i>length</i>. Oltre a questi attributi obbligatori, ci sono gli attributi facoltativi <i>digestType</i> e <i>digest</i>. Il valore dell'attributo <i>file</i> contiene l'URI del file (con codifica URL e se possibile riferito alla lobFolder più vicina) in cui è memorizzato il BLOB. Il valore dell'attributo <i>length</i> indica la lunghezza in UTF-8 e l'attributo facoltativo <i>digest</i> contiene l'informazione di integrità secondo l'attributo facoltativo <i>digestType</i>.</p> <p>Per il tipo di dati SQL:2008 DATALINK il <i>blobType</i> è usato con l'attributo <i>dlurlpathonly</i> (sezione 6.4 in ISO/IEC 9075-9:2008). È una restrizione di xs:anyURI secondo ISO/IEC 9075-9:2008, sezione 8 <i>URLs</i>. Le specifiche sono una traduzione diretta del formato degli URL HTTP e FILE specificati in [RFC 3986], ad eccezione di <i>localhost</i> che è stato omesso dal formato dell'URL FILE. <i>blobType</i> è definito in metadata.xsd.</p>	
P_4.3-4	Nei file degli schemi <code>table[numero].xsd</code> , i tipi di dati DISTINCT sono convertiti nel tipo di dati XML usato per rappresentare i loro tipi base.	O
P_4.3-5	<p>Nei file degli schemi <code>table[numero].xsd</code>, gli array sono convertiti in una sequenza di elementi XML strutturati <code><a1></code>, <code><a2></code> ecc., convertiti a loro volta nei tipi di dati XML corrispondenti al tipo base dell'array.</p> <p>Esempio Si veda l'esempio <code>table0.xsd</code> nell'allegato D.3c.</p>	O

ID	Descrizione requisito	O/F
P_4.3-6	<p>Nei file degli schemi <code>table[numero].xsd</code>, il tipo di dati definito dall'utente (<i>user-defined data type</i>, UDT) menzionato è convertito in una sequenza di elementi XML strutturati <code><a1></code>, <code><a2></code> ecc., convertiti a loro volta nei tipi di dati XML corrispondenti al tipo di ciascun attributo.</p> <p>Esempio Si veda l'esempio <code>table0.xsd</code> nell'allegato D.3c.</p>	O
P_4.3-7	<p>Nelle definizioni delle colonne in <code>metadata.xml</code>, le indicazioni nullable devono essere identiche a quelle del file corrispondente <code>table[numero].xsd</code>.</p> <p>Esempio</p>  <p>Nota La notazione SQL:2008 <code><nullable>true</nullable></code> diventa <code>minOccurs="0"</code> in XML. <code><nullable>false</nullable></code> corrisponde a <code>minOccurs="1"</code> in XML. Dato che è il valore predefinito, <code>minOccurs="1"</code> viene spesso ommesso. Se manca l'indicazione nullable significa che <code><nullable>true</nullable></code>.</p>	O
P_4.3-8	In <code>metadata.xml</code> , l'ordine delle colonne deve essere identico a quello nella tabella corrispondente <code>table[numero].xsd</code> .	O
P_4.3-9	Nella definizione della tabella in <code>metadata.xml</code> , l'ordine dei campi deve essere identico a quello della tabella corrispondente <code>table[numero].xsd</code> .	O

ID	Descrizione requisito	O/F
P_4.3-10	<p>In metadata.xml, il numero di righe di una tabella deve essere identico a quello nella tabella corrispondente table[numero].xsd.</p> <p>In metadata.xml, il numero di righe di una tabella deve rientrare nell'intervallo specificato nella tabella corrispondente table[numero].xsd.</p> <p>Esempio</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="432 517 724 1111" style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">metadata.xml</p> <pre> ... <table> <name>COUNTRIES</name> <folder>table0</folder> <description> <columns> <column> <column> <column> </columns> <primaryKey> <name>COUNTRY_C_ID_PK</name> <column>COUNTRY_ID</column> </primaryKey> <rows>25</rows> </table> ... </pre> </div> <div data-bbox="724 517 1362 898" style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center; margin: 0;">table0.xsd</p> <pre> <?xml version="1.0" encoding="utf-8" standalone="no" ?> <xs:schema xmlns="http://www.bar.admin.ch/xmns/siard/2/table.xsd" xmlns:xs="http://www ... <xs:element name="table"> <xs:complexType> ... <xs:sequence> <xs:element maxOccurs="unbounded" minOccurs="0" name="row" type="recordType"/> </xs:sequence> </xs:element> ... <xs:complexType name="recordType"> ... <xs:sequence> <xs:element name="c1" type="xs:string"/> <xs:element minOccurs="0" name="c2" type="xs:string"/> <xs:element minOccurs="0" name="c3" type="xs:decimal"/> </xs:sequence> </xs:complexType> ... </pre> </div> </div> <div data-bbox="724 898 1362 1111" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p style="text-align: center; margin: 0;">table0.xml</p> <pre> <?xml version="1.0" encoding="UTF-8" ?> <table xmlns="http://www.bar.admin.ch/xmns/siard/2/table.xsd" xmlns:xs="http://www.w3 <row><c1>AR</c1><c2>Argentina</c2><c3>2</c3></row> <row><c1>AU</c1><c2>Australia</c2><c3>3</c3></row> <row><c1>BE</c1><c2>Belgium</c2><c3>1</c3></row> ... <row><c1>US</c1><c2>United States of America</c2><c3>2</c3></row> <row><c1>ZM</c1><c2>Zambia</c2><c3>4</c3></row> <row><c1>ZW</c1><c2>Zimbabwe</c2><c3>4</c3></row> </table> </pre> </div>	

5 Requisiti per i metadati

Nell'archivio SIARD, i metadati memorizzano la struttura del database archiviato e indicano quali dati tabellari si trovano dove nell'archivio.

Tutti i metadati sono riuniti in un unico file denominato `metadata.xml` nella cartella `header/`. Il file ha una struttura gerarchica.

Per il file `metadata.xml` esiste la definizione di schema `metadata.xsd`, che è pure archiviata nella cartella `header/`.

ID	Descrizione requisito	O/F
M_5.0-1	La definizione di schema <code>metadata.xsd</code> è vincolante per il file <code>metadata.xml</code> , vale a dire che <code>metadata.xml</code> deve superare la validazione con lo schema <code>metadata.xsd</code> .	O

I contenuti dei singoli livelli sono definiti nel seguito.

5.1 Metadati a livello di database

Il file `metadata.xml` contiene le seguenti informazioni generali a livello di database:

ID	Descrizione requisito	O/F
M_5.1-1	Tutti i metadati definiti come obbligatori a livello di database in <code>metadata.xsd</code> devono essere compilati di conseguenza.	O

Nel file `metadata.xml` sono memorizzati i seguenti metadati del database:

Definizione	Significato	O/F
version	Versione del formato SIARD	O
dbname	Breve nome del database	O
description	Descrizione del significato e del contenuto del database nel suo complesso	F
archiver	Nome della persona che si è occupata dell'archiviazione dei dati tabellari provenienti dal database	F
archiverContact	Coordinate (telefono, e-mail) della persona che si è occupata dell'archiviazione dei dati tabellari provenienti dal database	F
dataOwner	Proprietario dei dati del database; istituzione o persona che, al momento dell'archiviazione, detiene il diritto di concedere i diritti di utilizzazione dei dati e che è responsabile del rispetto degli obblighi legali, per esempio delle linee guida sulla protezione dei dati	O
dataOriginTime span	Periodo di produzione dei dati del database; indicazione temporale approssimativa sotto forma di testo	O

Definizione	Significato	O/F
lobFolder	<p>URI di file che serve da URI di base per gli URI relativi che indicano la possibile collocazione di <i>large object</i> memorizzati esternamente. Se manca questo metadato, la cartella radice si trova per default nel file ZIP. Gli URI relativi delle lobFolder nei metadati delle colonne sono relativi a questo valore.</p> <p>Nota</p> <p>Se l'URI di file rimanda a un sistema di file esteso (<i>extended file system</i>) in cui i file ZIP sono trattati come cartelle, l'URI relativo rimanda alla cartella esterna in cui si trova il file SIARD. Se l'estensione del sistema di file non è supportata, devono essere usati URI di file assoluti per indicare la collocazione di file LOB memorizzati esternamente. Si raccomanda espressamente di rappresentare tutte le voci <i>lobFolder</i> in colonne e tutti gli attributi di file LOB sotto forma di URI relativi perché in caso di spostamento del file SIARD o del suo pacchetto informativo deve essere modificato soltanto l'URI generale per rinviare alla nuova collocazione.</p>	F
producerApplication	Nome e versione dell'applicazione usata per scaricare il file SIARD	F
archivalDate	Data di archiviazione; data dell'archiviazione dei dati tabellari	O
messageDigest	<p>Costituito dal <code>digestType</code> (MD5, SHA-1 o SHA-256) e dal corrispondente <code>digest</code>. Il <code>digest</code> è un buffer binario sotto forma di stringa di caratteri esadecimale o in alternativa – per SHA-1 o SHA-256 – di stringa base64. L'uso della codifica esadecimale o base64 è determinato dalla lunghezza del <code>digest</code> binario e della stringa di caratteri.</p> <p>Il <code>digest</code> è calcolato sulla base della cartella <code>content/</code>. È possibile memorizzare più codici <i>message digest</i> basati su diversi algoritmi. Se non è memorizzato alcun <i>message digest</i>, l'integrità deve essere assicurata memorizzando un codice analogo al di fuori del file SIARD¹⁴.</p> <p>Esempio</p> <p>Si veda l'esempio <code>metadata.xml</code> nell'allegato D.2.</p> <p>Raccomandazione</p> <p><i>Se si usa l'opzione <code>messageDigest</code>, è necessario fare quanto segue: i registri <code>content</code> e <code>header</code> devono essere memorizzati come voci separate (vuote) <code>content/</code> e <code>header/</code> nel file ZIP. Per poter verificare l'integrità dei dati primari, è necessario inserire la voce del registro <code>header</code> dopo tutti i dati primari e prima di tutti gli altri metadati nella voce <code>content/</code>. Il <code>messageDigest</code> menzionato sotto è calcolato partendo dall'offset 0 fino all'offset della voce <code>header/</code> del file SIARD.</i></p> <p><i>Il <code>messageDigest</code> indica valori esadecimali e quindi non è di stretta rilevanza che sia scritto in lettere maiuscole o minuscole. Sono tuttavia utilizzate e raccomandate per lo più le lettere minuscole (v. p. es. RFC 2831: https://www.ietf.org/rfc/rfc2831.txt).</i></p>	F

¹⁴ Un codice *message digest* memorizzato nel file SIARD non costituisce una garanzia d'integrità perché chi falsifica il file SIARD può falsificare anche questo codice. Soltanto la memorizzazione esterna del codice *message digest* permette di prevenire questo rischio. Può eventualmente risultare utile generare un codice *message digest* interno indipendente dai metadati al momento del download.

Definizione	Significato	O/F
clientMachine	Nome DNS del computer (client) su cui è stata eseguita l'archiviazione	F
databaseProduct	Prodotto e versione del database dal quale provengono i dati tabellari archiviati	F
connection	Stringa di connessione (<i>connection string</i>) usata per l'archiviazione dei dati tabellari	F
databaseUser	User ID nel database dell'utente dello strumento SIARD per l'archiviazione dei dati tabellari provenienti dal database	F
Schemi	Elenco degli schemi nel database	O
users	Elenco degli utenti del database	O
roles	Elenco dei ruoli del database	F
privileges	Elenco dei privilegi degli utenti e dei ruoli	F

5.2 Metadati a livello di schema

Come le informazioni generali sul database, anche i metadati degli schemi sono archiviati nel file `metadata.xml`.

ID	Descrizione requisito	O/F
M_5.2-1	Tutti i metadati definiti come obbligatori a livello di schema in <code>metadata.xsd</code> devono essere compilati di conseguenza.	O

Nel file `metadata.xml` sono memorizzati i seguenti metadati dello schema:

Definizione	Significato	O/F
name	Nome dello schema nel database	O
folder	Nome della cartella dello schema nel registro <code>content/</code> dell'archivio SIARD	O
types	Elenco dei tipi avanzati o strutturati (menzionati) nello schema	F
description	Descrizione del significato e del contenuto dello schema	F
tables	Elenco delle tabelle nello schema	F
views	Elenco delle viste memorizzate nello schema	F
routines	Elenco delle routine (precedentemente chiamate <i>stored procedures</i>) nello schema	F

5.3 Metadati a livello di tipo

ID	Descrizione requisito	O/F
M_5.3-1	I metadati dei tipi di uno schema possono essere memorizzati nel file <code>metadata.xml</code> .	F

Se viene archiviato un tipo avanzato o strutturato di dati, in `metadata.xml` sono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome del tipo di dati nello schema	O
category	Categoria del tipo avanzato o strutturato di dati (<i>distinct</i> o <i>udt</i>).	O
underSchema	Nome dello schema del supertipo, se il tipo di dati è basato su un supertipo	F
underType	Nome del supertipo, se il tipo di dati è basato su un supertipo	F
instantiable	Vero, se il tipo di dati può essere istanziato, altrimenti falso	O
final	Vero, se non può essere generato alcuno sottotipo per questo tipo di dati, altrimenti falso	O
base	Nome del tipo base (predefinito SQL), se la categoria è <i>distinct</i>	F
attributes	Elenco degli attributi, se la categoria è <i>udt</i>	F
description	Descrizione del significato e del contenuto del tipo di dati	F

5.4 Metadati a livello di attributo

ID	Descrizione requisito	O/F
M_5.4-1	Tutti i metadati del tipo <i>udt</i> definiti come obbligatori a livello di attributo in <code>metadata.xsd</code> devono essere compilati di conseguenza.	O

Nel file `metadata.xml` sono memorizzati i seguenti metadati degli attributi:

Definizione	Significato	O/F
name	Nome dell'attributo	O
type	Tipo predefinito di dati SQL:2008 dell'attributo secondo SQL:2008	F
typeOriginal	Tipo di colonna originale per il tipo di dati predefinito Nota Poiché i vari programmi di database che si definiscono conformi a SQL ammettono tipi di dati molto diversi, oltre al tipo di dati SQL:2008 viene indicato anche il tipo di colonna originale. Per ogni programma di database che supporta il formato SIARD deve essere definita e documentata la traduzione dei tipi di dati proprietari in tipi di dati SQL:2008 per l'applicazione corrispondente.	F
nullable	Elemento nullable dell'attributo Raccomandazione <i>Si raccomanda di non usare l'elemento nullable.</i>	F
typeSchema	Schema del tipo avanzato o strutturato di dati	F
typeName	Nome del tipo avanzato o strutturato di dati	F
defaultValue	Valore predefinito dell'attributo	F
description	Descrizione del significato e della funzione della routine	F
cardinality	Numero (massimo) di elementi, se l'attributo è un array	F

5.5 Metadati a livello di tabella

Come le informazioni generali sul database e i metadati degli schemi, anche i metadati a livello di tabella sono archiviati nel file `metadata.xml`.

ID	Descrizione requisito	O/F
M_5.5-1	Tutti i metadati definiti come obbligatori a livello di tabella in <code>metadata.xsd</code> devono essere compilati di conseguenza.	O

Nel file `metadata.xml` sono memorizzati i seguenti metadati delle tabelle:

Definizione	Significato	O/F
name	Nome della tabella nello schema	O
folder	Nome della cartella della tabella nella cartella dello schema	O
description	Descrizione del significato e del contenuto della tabella	F
columns	Elenco delle colonne della tabella	O
primaryKey	Chiave primaria della tabella	F
foreignKeys	Elenco delle chiavi esterne della tabella	F
candidateKeys	Elenco delle chiavi candidate della tabella	F
checkConstraints	Elenco dei vincoli della tabella	F
triggers	Elenco dei trigger della tabella	F
rows	Numero di record	O

5.6 Metadati a livello di colonna

Come le informazioni generali sul database, i metadati degli schemi e i metadati a livello di tabella, anche i metadati a livello di colonna sono archiviati nel file `metadata.xml`. I metadati delle colonne descrivono una colonna in una tabella o vista.

ID	Descrizione requisito	O/F
M_5.6-1	Tutti i metadati a livello di colonna definiti come obbligatori in <code>metadata.xsd</code> devono essere compilati di conseguenza.	O

Nel file `metadata.xml` sono memorizzati i seguenti metadati delle colonne:

Definizione	Significato	O/F
name	Nome della colonna nella tabella o vista Il nome della colonna deve essere univoco all'interno della stessa tabella.	O
lobFolder	Nome della cartella LOB sotto forma di URI di file relativo o assoluto, eventualmente nel sistema di file esterno. L'elemento può essere usato per la memorizzazione sia interna che esterna di <i>large object</i> . Esempio Si veda l'esempio <code>metadata.xml</code> nell'allegato D.2. Nota Questa voce è rilevante soltanto se la colonna è una colonna LOB (p. es. del tipo BLOB, CLOB, DATALINK o XML). Se manca, viene preso come valore predefinito ".", per esempio sotto forma di rimando alla stessa cartella della <code>lobFolder</code> a livello di database. Altrimenti il suo valore deve essere un URI di file (se possibile relativo), che indica la cartella in cui devono essere memorizzati i file di questa colonna LOB. Se il valore è un URI relativo, si assume che sia relativo alla voce <i>lobFolder</i> globale a livello di database. Gli attributi <i>file</i> relativi delle celle di questa colonna sono interpretati come relativi a questa cartella.	F
type	Tipo predefinito di colonna SQL:2008 Nota Se il tipo di dati della colonna è un tipo predefinito di dati, questo campo è obbligatorio. Altrimenti, il campo <i>typeName</i> deve rimandare a un tipo definito nell'elenco dei tipi.	O

Definizione	Significato	O/F
typeOriginal	Tipo di colonna originale Nota Poiché i vari programmi di database che si definiscono conformi a SQL ammettono tipi di dati molto diversi, oltre al tipo SQL:2008 viene indicato anche il tipo originale. Per ogni programma di database che supporta il formato SIARD deve essere definita e documentata la traduzione dei tipi proprietari in tipi SQL:2008 per l'applicazione corrispondente.	F
nullable	Voce facoltativa	F
typeSchema	Schema del tipo menzionato se la colonna non è un tipo predefinito di dati e se il tipo di dati menzionato non è definito nello stesso schema della tabella della colonna.	F
typeName	Nome del tipo avanzato o strutturato di dati di questa colonna	F
fields	Elenco dei campi nella colonna, se la colonna è un array o un tipo strutturato di dati della categoria <i>udt</i> .	F
defaultValue	Valore predefinito della colonna	F
contentType	Tipo MIME della colonna, se è una colonna BLOB e se tutte le voci della colonna contengono file dello stesso tipo MIME. Questo elemento puramente informativo aiuta a scegliere il visualizzatore corretto per gli oggetti binari. Può essere compilato manualmente o tramite il programma di download usando un meccanismo di riconoscimento del formato.	F
description	Descrizione del significato e del contenuto della colonna	F
cardinality	Numero (massimo) di elementi, se la colonna è un array	F

5.7 Metadati dei campi

ID	Descrizione dei requisiti	O/F
M_5.7-1	I metadati dei campi di una colonna o di un campo possono essere archiviati in <code>metadata.xml</code> .	F

Se la colonna o il campo è un array o un tipo avanzato o strutturato di dati della categoria *udt*, in `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	<p>Nome del campo nella colonna o nel campo</p> <p>Il nome del campo deve essere univoco all'interno della stessa colonna.</p> <p>Raccomandazione</p> <p><i>Per i contenitori (colonna o campo) del tipo <i>udt</i>, il nome del campo dovrebbe essere identico al nome dell'attributo corrispondente. Per i contenitori array, il nome del campo dovrebbe essere il nome del contenitore seguito dall'indice dell'array che inizia con 1 tra parentesi quadre. P. es. "Punto[1]", "Punto[2]" ecc.</i></p>	O
lobFolder	<p>Nome della cartella LOB sotto forma di URI di file relativo o assoluto, eventualmente nel sistema di file esterno. L'elemento può essere utilizzato per la memorizzazione sia interna che esterna di <i>large object</i>.</p> <p>Nota</p> <p>Questa voce è rilevante soltanto se il campo è un campo LOB (p. es. del tipo BLOB, CLOB, DATALINK o XML).</p> <p>Se manca, viene preso come valore predefinito ".", per esempio sotto forma di rimando alla stessa cartella della <i>lobFolder</i> a livello di database. Altrimenti il suo valore deve essere un URI di file (se possibile relativo), che indica la cartella in cui devono essere memorizzati i file di questo campo LOB.</p> <p>Se il valore è un URI relativo, si assume che sia relativo alla voce <i>lobFolder</i> globale a livello di database.</p> <p>Gli attributi <i>file</i> relativi delle celle della colonna sono interpretati come relativi a questa cartella.</p>	F
fields	Elenco dei campi nel campo, se il campo è un array o un tipo strutturato di dati della categoria <i>udt</i> .	F
mimeType	Tipo MIME del campo, se è un campo BLOB e se tutte le voci del campo contengono file dello stesso tipo MIME. Questo elemento puramente informativo aiuta a scegliere il visualizzatore corretto per gli oggetti binari. Può essere compilato manualmente o tramite il programma di download utilizzando un meccanismo di riconoscimento del formato.	F
description	Descrizione del significato e del contenuto del campo	F

5.8 Metadati della chiave primaria

Una chiave primaria è una chiave univoca (UNIQUE) usata per identificare un record.

ID	Descrizione requisito	O/F
M_5.8-1	I metadati della chiave primaria di una tabella possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una chiave primaria, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome della chiave primaria	O
column	Elenco delle colonne della chiave primaria	O
description	Descrizione del significato e del contenuto della chiave primaria	F

5.9 Metadati delle chiavi esterne

ID	Descrizione requisito	O/F
M_5.9-1	I metadati delle chiavi esterne di una tabella possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una chiave esterna, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome della chiave esterna	O
referencedSchema	Schema della tabella referenziata	O
referencedTable	Tabella referenziata Nota Il nome esterno referenziato della tabella può essere di tipo <i>table</i> o <i>schema.table</i> . Gli identificatori delimitati sono racchiusi tra virgolette.	O
reference	Referenza (elenco di colonne e colonne referenziate)	O
matchType	Tipo di corrispondenza (FULL, PARTIAL o SIMPLE)	F
deleteAction	Azione di cancellazione, p. es. CASCADE Nota Le azioni di cancellazione e di modifica comprendono le azioni consentite dallo standard SQL:2008.	F
updateAction	Azione di modifica, p. es. SET DEFAULT	F

Definizione	Significato	O/F
description	Descrizione del significato e del contenuto della chiave esterna	F

5.10 Metadati delle referenze

ID	Descrizione requisito	O/F
M_5.10-1	I metadati delle referenze usate per la chiave esterna possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una chiave esterna, nel file `metadata.xml` sono memorizzati i seguenti metadati:

Definizione	Significato	O/F
column	Nome della colonna	O
referenced	Nome della colonna referenziata	O

5.11 Metadati della chiave candidata

Le chiavi candidate sono chiavi univoche (UNIQUE) perché possono entrare in considerazione come candidate per la chiave primaria. In `metadata.xsd`, sia la chiave primaria sia la chiave candidata sono del tipo `uniqueKeyType`. I requisiti per la chiave candidata sono pertanto identici a quelli per la chiave primaria (M_5.8-1).

ID	Descrizione requisito	O/F
M_5.11-1	I metadati della chiave candidata di una tabella possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una chiave candidata, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome della chiave candidata	O
column	Elenco delle colonne della chiave candidata	O
description	Descrizione del significato e del contenuto della chiave candidata	F

5.12 Metadati del vincolo di controllo

Il vincolo di controllo consiste in una condizione da verificare. Questa è specificata come espressione del tipo `BOOLEAN` (con valore `true`, `false` o `unknown`) nella sintassi SQL:2008.

ID	Descrizione requisito	O/F
M_5.12-1	I metadati del vincolo di controllo di una tabella possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviato un vincolo di controllo, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome del vincolo di controllo	O
condition	Condizione del vincolo di controllo	O
description	Descrizione del significato e del contenuto del vincolo di controllo	F

5.13 Metadati a livello di trigger

ID	Descrizione requisito	O/F
M_5.13-1	I metadati del trigger di una tabella possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviato un trigger, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome del trigger nella tabella	O
actionTime	BEFORE, AFTER o INSTEAD OF	O
triggerEvent	INSERT, DELETE, UPDATE [OF <trigger column list>]	O
aliasList	<old or new value alias list>	F
triggeredAction	<triggered action>	O
description	Descrizione del significato e del contenuto del trigger	F

5.14 Metadati a livello di vista

ID	Descrizione requisito	O/F
M_5.14-1	I metadati della vista di uno schema possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una vista, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome della vista nello schema	O
columns	Elenco dei nomi delle colonne della vista Nota I metadati delle colonne di una vista sono strutturati in modo identico a quelli di una tabella.	O
query	Query SQL:2008 che definisce la vista	F
queryOriginal	Query SQL originale che definisce la vista Nota Poiché i vari programmi di database che si definiscono conformi a SQL ammettono sintassi di query molto diverse, oltre alla query SQL:2008 viene indicata anche la query originale. Per ogni programma di database che supporta il formato SIARD deve essere definita e documentata la traduzione della sintassi di query proprietaria in tipi SQL:2008 per l'applicazione corrispondente.	F
rows	Numero di record	F
description	Descrizione del significato e del contenuto della vista	F

5.15 Metadati a livello di routine

ID	Descrizione requisito	O/F
M_5.15-1	I metadati della routine di uno schema possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una routine, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
specificName	Nome specifico che identifica in modo univoco la routine nello schema ¹⁵ .	O
name	Nome della routine nello schema	O
description	Descrizione del significato e del contenuto della routine	F

¹⁵ Con l'introduzione di elementi orientati agli oggetti in SQL:1999 è diventato possibile anche il «sovraccarico», che permette di dare lo stesso nome a due diverse routine (procedure o funzioni), a condizione che abbiano un elenco diverso di parametri. È pertanto necessario rinunciare al requisito dell'univocità del nome di una routine nello schema. In cambio è stato introdotto il «nome specifico» che identifica in modo univoco la routine nello schema.

Definizione	Significato	O/F
source	Codice sorgente originale della routine (VBA, PL/SQL, JAVA) Nota Poiché molti programmi di database hanno routine proprietarie che non possono essere trasformate in una query conforme a SQL:2008, può essere archiviato il codice sorgente originale della routine (p. es. PL/SQL per i database Oracle, VBA per i moduli MS Access).	F
body	Codice sorgente della routine conforme a SQL:2008	F
characteristic	Caratteristica della routine	F
returnType	Tipo di ritorno della routine (se si tratta di una funzione)	F
parameters	Elenco dei parametri	F

5.16 Metadati dei parametri

ID	Descrizione requisito	O/F
M_5.16-1	I metadati dei parametri usati per la routine possono essere archiviati nel file <code>metadata.xml</code> .	F

Se è archiviata una routine, nel file `metadata.xml` vengono memorizzati i seguenti metadati:

Definizione	Significato	O/F
name	Nome del parametro	O
mode	Modalità del parametro (IN, OUT o INOUT)	O
type	Tipo predefinito di parametro SQL:2008 Nota Questo campo deve essere usato se il tipo di dati della colonna è un tipo predefinito di dati. Altrimenti, il campo <i>typeName</i> deve rimandare a un tipo definito nell'elenco dei tipi.	O
typeOriginal	Tipo di parametro originale Nota Poiché i vari programmi di database che si definiscono conformi a SQL ammettono tipi di dati molto diversi, oltre al tipo di dati SQL:2008 viene indicato anche il tipo di colonna originale. Per ogni programma di database che supporta il formato SIARD deve essere definita e documentata la traduzione dei tipi di dati proprietari in tipi di dati SQL:2008 per l'applicazione corrispondente.	F
typeSchema	Schema del tipo menzionato, se il parametro non è un tipo predefinito di dati e se il tipo di dati menzionato non è definito nello stesso schema della tabella della colonna.	F

Definizione	Significato	O/F
typeName	Nome del tipo avanzato o strutturato di dati di questo parametro	F
description	Descrizione del significato e della funzione della routine	F
cardinality	Numero (massimo) di elementi, se il parametro è un array	F

5.17 Metadati a livello di utente

ID	Descrizione requisito	O/F
M_5.17-1	I metadati degli utenti possono essere archiviati nel file <code>metadata.xml</code> .	F

Nel file `metadata.xml` sono memorizzati i seguenti metadati degli utenti:

Definizione	Significato	O/F
name	Nome dell'utente	O
description	Descrizione del significato e della funzione dell'utente	F

5.18 Metadati a livello di ruolo

ID	Descrizione requisito	O/F
M_5.18-1	I metadati dei ruoli possono essere archiviati nel file <code>metadata.xml</code> .	F

Nel file `metadata.xml` sono memorizzati i seguenti metadati dei ruoli:

Definizione	Significato	O/F
name	Nome del ruolo	O
admin	Amministratore del ruolo (utente o ruolo)	O
description	Descrizione del significato e della funzione del ruolo	F

5.19 Metadati a livello di privilegi

ID	Descrizione requisito	O/F
M_5.19-1	I metadati dei privilegi possono essere archiviati nel file <code>metadata.xml</code> .	F

Nel file `metadata.xml` sono memorizzati i seguenti metadati dei privilegi:

Definizione	Significato	O/F
type	Privilegio accordato (p. es. SELECT)	O
object	Oggetto cui si applica il privilegio	F
grantor	Persona autorizzata a concedere il privilegio	O
grantee	Beneficiario del privilegio (utente o ruolo)	O
option	Opzione GRANT (ADMIN o GRANT)	F
description	Descrizione del significato e della funzione del privilegio	F

6 Requisiti per i dati tabellari

Come già descritto, i dati tabellari di un database relazionale archiviato si trovano nella cartella `content/` nel documento radice dell'archivio SIARD. Sono collocati nella cartella dello schema o della tabella in base al tipo di dati.

I dati tabellari sono sempre memorizzati in un file XML. Per ogni tabella è generata una definizione di schema XML che specifica il formato XML per la memorizzazione dei dati tabellari. Di conseguenza, per ogni tabella è generato un file `table[numero].xml` per la definizione dello schema `table[numero].xsd`.

ID	Descrizione requisito	O/F
T_6.0-1	Tutti i dati tabellari (dati primari) devono soddisfare i requisiti di consistenza di SQL:2008. Un file SIARD che supera la validazione sintattica con i vari XSD, ma che viola semanticamente lo standard SQL non è conforme alle presenti specifiche. In particolare, i valori delle tabelle devono soddisfare i vincoli dei tipi SQL nei metadati. Devono inoltre essere soddisfatte tutte le condizioni memorizzate nei metadati delle chiavi primarie, candidate ed esterne e le condizioni nullable.	O
T_6.0-2	La definizione di schema <code>table[numero].xsd</code> è vincolante per il file <code>table[numero].xml</code> , vale a dire che <code>table[numero].xml</code> deve superare la validazione con lo schema <code>table[numero].xsd</code> .	O

6.1 Definizione dello schema delle tabelle

Il file `table[numero].xsd` contiene le seguenti definizioni dello schema di una tabella:

ID	Descrizione requisito	O/F
T_6.1-1	Ogni tabella deve avere una definizione di schema XML che specifichi il formato XML per la memorizzazione dei dati tabellari.	O
T_6.1-2	Questa definizione riflette lo schema SQL dei metadati della tabella e specifica che la tabella è memorizzata sotto forma di sequenza di righe che contengono una sequenza di voci di colonna con diversi tipi XML. Il nome del tag della tabella è <i>table</i> , il nome del tag del record è <i>row</i> , i tag delle colonne sono <i>c1</i> , <i>c2</i> ecc. I tag delle colonne iniziano sempre con <i>c1</i> e aumentano progressivamente di 1. Non possono quindi esserci interruzioni nella numerazione, perché in caso di valore NULL verrebbe a mancare la colonna corrispondente nel file XML. Esempio Si veda l'esempio <code>table2.xsd</code> nell'allegato D.3a.	O
T_6.1-3	La mappatura dei tipi da usare nelle definizioni degli schemi delle tabelle è specificata in P_4.3-3. Oltre ai tipi standard di schema XML, vengono usati i seguenti tipi speciali: <code>clobType</code> , <code>blobType</code> , <code>datalinkType</code> , <code>dateType</code> , <code>timeType</code> , <code>dateTimeType</code>	O

ID	Descrizione requisito	O/F
T_6.1-4	<p>Valori multipli di tipi avanzati o strutturati devono essere memorizzati come elementi separati all'interno dei tag delle celle.</p> <p>I nomi dei singoli elementi di un array sono a1, a2 ecc.</p> <p>I nomi dei singoli elementi di un <i>udt</i> sono u1, u2 ecc.</p> <p>I nomi iniziano sempre con a1 o u1 e aumentano progressivamente di 1. Non possono quindi esserci interruzioni nella numerazione, perché in caso di valore NULL verrebbe a mancare la colonna corrispondente nel file XML.</p> <p>Esempio</p> <p>Si veda l'esempio <code>table0.xsd</code> nell'allegato D.3c.</p>	O

6.2 Celle di dati *large object*

ID	Descrizione requisito	O/F
T_6.2-1	<p>I <i>large object</i> possono essere memorizzati inline nel file <code>table[numero].xml</code>, sotto forma di voci di file separate nel file SIARD (interno) o sotto forma di file separati nel sistema di file esterno al file SIARD.</p>	O

Se i *large object* sono memorizzati sotto forma di file separati (interni o esterni), gli attributi *file* e *length* devono essere obbligatoriamente memorizzati in una cella LOB del file `table[numero].xml`. Se i *large object* sono memorizzati inline, questi attributi sono facoltativi:

Definizione	Significato	O/F
file	Se il <i>large object</i> è memorizzato separatamente, questo elemento indica la collocazione e il nome del file nella cella o nell'attributo della cella sotto forma di URI di file. Se è relativo, l'URI è considerato relativo alla <i>lobFolder</i> (della colonna o dell'attributo) dell'elemento che la racchiude.	O ¹⁶
length	Lunghezza (per BLOB e DATALINK in byte, per CLOB e XML in caratteri)	O ¹⁶
digestType	<p>Contiene il tipo dell'informazione di integrità (digest): MD5, SHA-1 o SHA-256.</p> <p>Raccomandazione</p> <p>Si raccomanda d'impostare l'attributo (in combinazione con il digest) per tutti i LOB memorizzati in file separati.</p>	F
digest	<p>Informazione di integrità sul <i>large object</i>.</p> <p>Raccomandazione</p> <p>Si raccomanda d'impostare questo attributo (in combinazione con il digestType) per tutti i LOB memorizzati in file separati.</p>	F

¹⁶ F se il *large object* è memorizzato inline in `table[numero].xml`.

Definizione	Significato	O/F
dlurlpathonly	Il percorso (compreso il nome) del riferimento del file indicato nel sistema di gestione del database relazionale da cui il LOB è stato esportato in SIARD (ISO/IEC 9075-9:2008 6.4 <i>String value function</i>). Si applica soltanto ai LOB esterni.	

6.3 Celle di dati *data* e *timestamp*

ID	Descrizione requisito	O/F
T_6.3-1	Le date e i timestamp devono essere limitati agli anni 0001-9999, in conformità alle specifiche SQL:2008. Questa restrizione è imposta nelle definizioni di <i>dateType</i> e <i>dateTimeType</i> .	O
T_6.3-2	Le date, le ore e timestamp devono essere memorizzati in UTC eventualmente con la terminazione Z. Questa restrizione è imposta nelle definizioni <i>dateType</i> , <i>timeType</i> e <i>dateTimeType</i> . Raccomandazione <i>Memorizzare tutte le date, le ore e i timestamp con la terminazione Z.</i>	O

6.4 Dati tabellari

Il file `table[numero].xml` contiene i dati tabellari della tabella:

ID	Descrizione requisito	O/F
T_6.4-1	I dati tabellari di ogni tabella devono essere memorizzati in un file XML.	O
T_6.4-2	Il file <i>table</i> è composto da elementi <i>row</i> che suddividono i dati di una riga in diverse colonne (c1, c2 ecc.). Esempio Si veda l'esempio <code>table2.xml</code> nell'allegato D.4a.	O
T_6.4-3	Se è NULL, la cella di una colonna o di un campo deve essere omessa. Se è uguale a "" (stringa di lunghezza 0), deve essere presente ma vuota.	O
T_6.4-4	Se contiene un valore complesso (<i>array</i> , <i>udt</i>), la cella di una colonna è rappresentata da una sequenza di sottoelementi della cella (a1, a2 ecc. per <i>array</i> , u1, u2 ecc. per <i>udt</i>), che a loro volta contengono i valori corrispondenti. Questi valori possono a loro volta essere complessi. Esempio Si veda l'esempio <code>table0.xml</code> nell'allegato D.4c.	F

ID	Descrizione requisito	O/F
T_6.4-5	<p>Se una tabella contiene dati del tipo <i>large object</i> (BLOB, CLOB, DATALINK o XML ecc.), possono essere creati file separati e invece del contenuto delle celle può essere memorizzata la collocazione del file.</p> <p>La decisione di memorizzare <i>large object</i> in file separati anziché inline è presa dal software che genera il file SIARD.</p> <p>Per evitare che siano generate cartelle vuote, le cartelle vengono create soltanto se necessarie (cioè se contengono dati).</p> <p>Se i <i>large object</i> sono memorizzati in un file separato, è obbligatorio impostare gli attributi <i>file</i> e <i>length</i>. Il valore dell'attributo <i>file</i> contiene l'URI del file (con codifica URL e se possibile riferito alla lobFolder più vicina) in cui è memorizzato il LOB. Il valore dell'attributo <i>length</i> è la lunghezza (per BLOB e DATALINK in byte, per CLOB e XML in caratteri). Il <i>digest</i> registra un <i>message digest</i> sul file LOB per permettere di controllare l'integrità dell'archivio SIARD anche quando alcuni LOB sono memorizzati esternamente.</p> <p>Esempio Si veda l'esempio <code>table7.xml</code> nell'allegato D.4b.</p> <p>Raccomandazione <i>Si raccomanda espressamente di salvare inline in una colonna o tutti i large object o nessuno.</i> <i>Si raccomanda di uniformare le cartelle e i file LOB e di usare, al posto del nome originale, per esempio <code>lob4/ e record0.bin o record0.txt</code>.</i></p>	O

7 Requisiti per la struttura delle cartelle per LOB memorizzati al di fuori del file SIARD

I seguenti requisiti si applicano alla struttura delle cartelle per LOB memorizzati al di fuori del file SIARD.

I LOB possono essere memorizzati al di fuori del file SIARD, a prescindere che siano interni (p. es. dati del tipo BLOB) o esterni (p. es. dati del tipo DATALINK).

I LOB sono memorizzati come file in cartelle, una per ogni schema e una per ogni colonna.

Le cartelle sono specificate nel valore dell'elemento *lobFolder* a livello di *siardArchive* (p. es. per tutti gli schemi) e a livello di colonna (p. es. per ogni singola colonna).

I LOB esterni sono dati memorizzati al di fuori del database relazionale, in conformità alla parte 9 di SQL:2008 (SQL/MED).

Secondo lo standard ISO/IEC 9075-9:2008 (SQL/MED), è ammesso soltanto un collegamento tra una cella di tipo DATALINK e un file esterno, altrimenti viene generato il messaggio d'errore «datalink exception — external file already linked»¹⁷.

7.1 Struttura delle cartelle per LOB memorizzati al di fuori del file SIARD

ID	Descrizione requisito	O/F
L_7.1-0	<p>I LOB sono memorizzati come file in cartelle, una per ogni schema, tabella e colonna.</p> <p>Le cartelle sono definite nel file <code>metadata.xml</code> in base al valore dei rispettivi elementi <code><siardArchive/></code> <code><lobFolder/></code> e <code><column/></code> <code><lobFolder/></code>.</p> <p>Il nome del file LOB deve essere uniformato (in conformità a P_4.2-3) come definito di seguito.</p> <p>Alla struttura della cartella e al nome delle cartelle e dei file si applica quanto segue:</p> <ul style="list-style-type: none"> • una cartella LOB principale denominata <code>[databaseName]_lobs</code>; • una cartella LOB per ogni colonna denominata in base allo schema <i>i</i>, alla tabella <i>j</i>, alla colonna <i>k</i>; p. es.: <code>s[i]_t[j]_c[k]</code>; • una cartella denominata <code>seg_0</code>; • un file LOB denominato in base alla tabella <i>j</i>, alla colonna <i>k</i> e alla riga <i>l</i> del LOB; p. es. <code>t[j]_c[k]_r[l]</code>; • un suffisso del nome del file LOB denominato <code>bin</code> (o un'estensione di file associata al tipo MIME del file LOB, nel caso in cui questo sia noto [v. restrizioni P_4.2-6]). 	O

¹⁷ Si veda SQL:2008, parte 9, 15.2 *Effect of inserting tables into base tables*, 1, b, ii).

La soluzione conforme allo standard per gestire più riferimenti a uno stesso file è quindi quella di creare una relazione «uno a molti» tra la cella di tipo DATALINK e le celle che si riferiscono al file e non quella di creare diverse celle di tipo DATALINK che fanno riferimento allo stesso file.

ID	Descrizione requisito	O/F
	<p>Nota Il valore di base delle tabelle è 0, quello delle colonne e delle righe 1.</p> <p>Esempio Northwind.siard Northwind_lobs/ s0_t2_c4/ seg_0/ t2_c4_r1.bin t2_c4_r2.bin t2_c4_r3.bin t2_c4_r4.bin t2_c4_r5.bin t2_c4_r6.bin t2_c4_r7.bin t2_c4_r8.bin s0_t2_c8/ <!-- new column --> seg_0/ t2_c8_r3.bin s0_t11_c6/ <!-- new column --> seg_0/ t11_c6_r7.bin</p> <p>metadata.xml <?xml version="1.0" encoding="UTF-8" standalone="yes"?>... <siardArchive>...<lobFolder>./Northwind_lobs/</lobFolder> .. <column>...<lobFolder>s0_t2_c4/</lobFolder>...</column> <column>...<lobFolder>s0_t2_c8/</lobFolder>...</column> .. <column>...<lobFolder>s0_t11_c6/</lobFolder>...</column></p> <p>table2.xml <row><c1>1</c1><c2>Beverages</c2><c3>Soft drinks, coffees, teas, beers, and ales</c3> <c4 file="seg_0/t2_c4_r1.bin" length="10151" /></row> <row><c1>5</c1><c2>Seafood</c2><c3></c3><c4 file="seg_0/t2_c4_r5.bin" ... /></row> <row><c1>8</c1><c2>Candy</c2><c3></c3><c4 file="seg_0/t2_c4_r8.bin" ... /></row></p>	
L_7.1-1	Le cartelle [databaseName]_lobs/seg_[]/ possono anche essere impacchettate in un file ZIP e avere il suffisso .zip.	F

8 Problemi di scalabilità

Possono verificarsi problemi di scalabilità quando grandi quantità di LOB o LOB di grandi dimensioni vengono memorizzati al di fuori del file SIARD. In rari casi questi problemi possono insorgere anche con il file SIARD stesso. In genere dipendono dall'implementazione, ma milioni di LOB o LOB di diverse TB possono costituire una sfida.

Nel presente capitolo è illustrato come gestire i problemi di scalabilità per migliorare l'interoperabilità.

8.1 Segmentazione di LOB memorizzati al di fuori del file SIARD

Per trattare in modo efficiente e semplice file e cartelle (p. es. copia, hashing, convalida ecc.) è possibile limitare il numero e le dimensioni dei LOB in una cartella e suddividere i LOB in più cartelle. Il numero e le dimensioni massime dei LOB dipendono dall'implementazione.

ID	Descrizione requisito	O/F
S_8.1-0	<p>Per la struttura e il nome delle cartelle per la segmentazione dei LOB si veda L_7.1-0.</p> <p>Alla struttura e al nome delle cartelle si applica quanto segue:</p> <ul style="list-style-type: none"> una nuova cartella denominata <code>seg_[s]</code> se è raggiunto il numero massimo di file per cartella; una nuova cartella denominata <code>seg_[s]</code> se sono raggiunte le dimensioni massime dei file per cartella. <p>Nota</p> <p>Il valore di base delle tabelle è 0, quello delle colonne e delle righe 1. Il numero e le dimensioni massime dipendono dall'implementazione.</p> <p>Esempio</p> <pre>Northwind.siard Northwind_lobs/ s0_t2_c4/ seg_0/ t2_c4_r1.bin t2_c4_r2.bin t2_c4_r3.bin t2_c4_r4.bin seg_1/ <!-- folder file number limit (4) --> t2_c4_r5.bin t2_c4_r6.bin t2_c4_r7.bin seg_2/ <!-- folder file size limit (8 GB) --> t2_c4_r8.bin s0_t2_c8/ seg_0/ t2_c8_r3.bin s0_t11_c6/</pre>	O ¹⁸

¹⁸ Obbligatorio soltanto se i LOB sono segmentati.

ID	Descrizione requisito	O/F
	<pre> seg_0/ t11_c6_r7.bin metadata.xml <?xml version="1.0" encoding="UTF-8" standalone="yes"?>... <siardArchive>...<lobFolder>/Northwind_lobs/</lobFolder> .. <column>...<lobFolder>s0_t2_c4/</lobFolder>...</column> <column>...<lobFolder>s0_t2_c8/</lobFolder>...</column> .. <column>...<lobFolder>s0_t11_c6/</lobFolder>...</column> table2.xml <row><c1>1</c1><c2>Beverages</c2><c3>Soft drinks, coffees, teas, beers, and ales</c3> <c4 file="seg_0/t2_c4_r1.bin" length="10151" /></row> <row><c1>5</c1><c2>Seafood</c2><c3></c3><c4 file="seg_1/t2_c4_r5.bin" ... /></row> <row><c1>8</c1><c2>Candy</c2><c3></c3><c4 file="seg_2/t2_c4_r8.bin" ... /></row> </pre>	

8.1.1 Suddivisione di LOB in parti binarie

Può succedere che un LOB superi le dimensioni massime previste per una cartella (p. es. nel caso di un film lungo). In questo caso è necessario procedere a una suddivisione binaria del file stesso per ridurlo in parti più piccole utilizzando una convenzione di denominazione.

ID	Descrizione requisito	O/F
S_8.1.1-0	<p>Se supera le dimensioni massime previste per la cartella, un LOB deve essere ridotto in parti di file mediante una suddivisione binaria.</p> <p>Ogni parte del file deve avere il suffisso <code>_part[nnn]</code>, con nnn che inizia da 001.</p> <p>Le parti del file devono essere memorizzate in ordine.</p> <p>Esempio</p> <pre> Northwind.siard Northwind_lobs/ s0_t2_c4/ seg_0/ t2_c4_r1.bin t2_c4_r2.bin t2_c4_r3.bin t2_c4_r4.bin seg_1/ <!-- folder file number limit (4) --> t2_c4_r5.bin t2_c4_r6.bin_part001 <!-- file size limit (8 GB) --> seg_2/ t2_c4_r6.bin_part002 t2_c4_r7.bin </pre>	O ¹⁹

¹⁹ Obbligatorio soltanto se i LOB sono suddivisi.

ID	Descrizione requisito	O/F
	<pre> t2_c4_r8.bin s0_t2_c8/ seg_0/ t2_c8_r3.bin s0_t11_c6/ seg_0/ t11_c6_r7.bin </pre>	

8.1.2 File di mappatura delle cartelle dei segmenti

In caso di problemi di scalabilità dovuti a grandi quantità di LOB, la segmentazione in cartelle potrebbe non essere sufficiente perché i LOB si trovano spesso nella stessa colonna (p. es. in una tabella le immagini delle patenti di guida di 5 milioni di persone si trovano in un'unica colonna con 5 milioni di righe contenente ciascuna il LOB dell'immagine di una patente).

Di conseguenza, avere un'unica cartella (percorso) per una colonna può non essere sufficiente, in particolare durante la creazione del file SIARD, il trasferimento o la creazione di un pacchetto SIP.

In questo caso si può usare in via facoltativa un file di mappatura. Da notare che la mappatura è rilevante soltanto al momento della creazione del file. Durante il trasferimento o dopo la ricezione in un archivio potrebbe essere necessario rivederla.

ID	Descrizione requisito	O/F
S_8.1.2-0	<p>Se è usato in via facoltativa, un file di mappatura deve essere denominato <code>mapping.txt</code>.</p> <p>Il file <code>mapping.txt</code> deve trovarsi allo stesso livello della cartella del file SIARD.</p> <p>Il file <code>mapping.txt</code> deve contenere il percorso alla cartella segmentata LOB seguito da uno spazio e dall'URI della collocazione della cartella segmentata (URI in conformità a RFC 3986).</p> <p>Esempio</p> <pre> Northwind_lobs/s0_t2_c4/seg_0/ file://storagesrv1.sfa.ch/Home/stor/ Northwind_lobs/s0_t2_c4/seg_1/ file://storagesrv1.sfa.ch/Home/stor/ Northwind_lobs/s0_t2_c4/seg_2/ file://storagesrv2.sfa.ch/Home/stor/ </pre> <p>Esempio di mappatura della cartella <code>s0_t2_c4/seg_0/</code>:</p> <pre> file://storagesrv1.sfa.ch/Home/stor/Northwind_lobs/s0_t2_c4/seg_0/ </pre>	O

8.1.3 File MANIFEST per LOB memorizzati al di fuori del file SIARD

Per LOB memorizzati al di fuori del file SIARD può essere creato in via facoltativa un file MANIFEST per facilitarne la gestione, in particolare durante la creazione del file SIARD, il trasferimento o la creazione di un pacchetto SIP.

Lo scopo di specificare il formato del file MANIFEST è di migliorare l'interoperabilità.

Le informazioni contenute nel file MANIFEST sono già disponibili in altre parti del file SIARD e devono basarsi su queste. Il file MANIFEST ha quindi soltanto un'utilità pratica.

Le informazioni sulla collocazione di un LOB memorizzato al fuori del file SIARD sono deducibili dal percorso combinato del valore dell'elemento *lobFolder* del database e della colonna nel file `metadata.xml` e del valore dell'attributo del file dell'elemento *c[c]* della colonna nel file `table[t].xml`.

Le informazioni sul valore di *digest* sono deducibili dal valore dell'attributo *digest* (v. T_6.4-5).

ID	Descrizione requisito	O/F
S_8.1.3-0	<p>Se lo si utilizza per LOB memorizzati al di fuori del file SIARD, il file MANIFEST deve essere conforme alla struttura basata su <i>GNU md5sum-invocation</i>, a prescindere dall'algoritmo di <i>checksum</i> (somma di controllo).</p> <p>https://www.gnu.org/software/coreutils/manual/coreutils.html#md5sum-invocation https://www.gnu.org/software/coreutils/manual/coreutils.html#sha2-utilities</p> <p>Somma di controllo, spazio, bandiera che indica il modo di inserimento (binario o testo) e nome del file.</p> <p>Il modo binario è indicato con '*', il modo testo con ' ' (spazio).</p> <p>Esempio</p> <pre>2de1ac4c4e8ebb853e17db01af3fb7c3 */Northwind_lobs/s0_t2_c4/seg_2/t2_c4_r8.bin</pre>	O

8.2 Suddivisione del file SIARD in parti binarie

In rari casi può succedere che un file SIARD superi le dimensioni massime previste dipendenti dall'implementazione, per esempio a causa di un numero massimo di cartelle poco elevato combinato a dimensioni totali molto elevate delle tabelle.

In questi casi è necessario procedere a una suddivisione binaria del file stesso per ridurlo in parti più piccole utilizzando una convenzione di denominazione.

Lo scopo di specificare il formato della suddivisione è migliorare l'interoperabilità.

ID	Descrizione requisito	O/F
S_8.2-0	<p>Se supera le dimensioni massime previste, un file SIARD deve essere ridotto in parti procedendo a una suddivisione binaria.</p> <p>Ogni parte del file deve avere il suffisso <code>_part[nnn]</code>, con nnn che inizia da 001.</p> <p>Le parti del file devono essere memorizzate in ordine.</p> <p>Esempio</p> <pre>Northwind.siard_part001 <!-- file size limit --> Northwind.siard_part002 Northwind.siard_part003</pre>	O ²⁰

²⁰ Obbligatorio soltanto se il SIARD file è suddiviso.

ID	Descrizione requisito	O/F
	Il numero e le dimensioni massime dipendono dall'implementazione.	

9 Versione e validità delle specifiche

Le specifiche sono disponibili nella versione 2.2.

10 Processo di gestione delle modifiche

Il processo di modifica di questo standard è gestito da DILCIS.

11 Esclusione della responsabilità e diritti di terzi

DILCIS e l'AFS non rispondono in alcun caso per decisioni prese o misure adottate dall'utente sulla base del presente documento. DILCIS e l'AFS non possono fornire alcuna assicurazione o garanzia che le informazioni e i documenti messi a disposizione siano aggiornati, completi, corretti e privi di errori. È esclusa, nella misura consentita dalla legge, qualsiasi responsabilità per danni subiti dall'utente dall'uso del presente standard.

12 Diritti d'autore

Il presente standard è di proprietà intellettuale dei suoi autori che s'impegnano a metterlo a disposizione gratuitamente per uso illimitato e per l'ulteriore sviluppo.

Allegato A – Collaborazione e verifica

Autori

Krystyna Ohnesorge, Archivio federale svizzero, krystyna.ohnesorge@bar.admin.ch

Thomas Hartwig, Enter AG, hartwig.thomas@enterag.ch

Andreas Voss †, Archivio federale svizzero

Marcel Büchler, Archivio federale svizzero, marcel.buechler@bar.admin.ch

Audun Lund, Archivio federale svizzero, audun.lund@bar.admin.ch

Claire Röthlisberger-Jourdan, KOST, claire.roethlisberger@kost.admin.ch

Anders Bo Nielsen, Archivio nazionale danese, abn@sa.dk

Arne-Kristian Groven, Archivio nazionale norvegese, arngro@arkivverket.no

Luis Faria, KEEP SOLUTIONS, LDA, lfaria@keep.pt

Collaboratori

Hedi Bruggisser, Archivio di Stato del Cantone di Turgovia, hedi.bruggisser@tg.ch

Georg Büchler, KOST, georg.buechler@kost.admin.ch

Boris Domajnko, Archivio nazionale sloveno, boris.domajnko@gov.si

Alain Dubois, , Archivio di Stato del Cantone del Vallese, alain.dubois@admin.vs.ch

Bruno Ferreira, KEEP SOLUTIONS, LDA

Miguel Guimarães, KEEP SOLUTIONS, LDA

Martin Kaiser, KOST, martin.kaiser@kost.admin.ch

Lambert Kansy, Archivio di Stato del Cantone di Basilea Città, lambert.kansy@bs.ch

Markus Lischer Archivio di Stato del Cantone di Lucerna, markus.lischer@lu.ch

Zoltán Lux, Archivio nazionale ungherese, lux.zoltan@mnl.gov.hu

Rebekka Plüss, Archivio di Stato del Cantone di Zurigo, rebekka.pluess@ji.zh.ch

Lauri Rätsep, Archivio nazionale estone, lauri.ratsep@ra.ee

Hélder Silva, KEEP SOLUTIONS, LDA, hsilva@keep.pt

Mario Spuler, Fachlabor Gubler, m.spuler@fachlabor-gubler.ch

Martin Dew-Hattens, Archivio nazionale danese, mdh@sa.dk

Allegato B – Abbreviazioni e glossario

Termine	Descrizione
AIP	Pacchetto di archiviazione (<i>Archival Information Package</i>): secondo il modello OAIS, il pacchetto AIP è derivato dal pacchetto SIP durante il processo di archiviazione dei documenti digitali. L'AIP è la forma di pacchetto informativo in cui sono conservati i documenti digitali nel magazzino digitale.
Archiviazione permanente o a lungo termine	Termine per indicare la conservazione e la disponibilità a tempo indeterminato di informazioni digitali. Oltre alla preservazione della sequenza di bit (<i>bit stream</i>) delle informazioni archiviate, include anche la garanzia dell'interpretabilità e presentazione dei dati in una forma leggibile e comprensibile per l'essere umano.
Archivio	<ol style="list-style-type: none"> 1. Istituzione o servizio che valuta, conserva e rende accessibili i fondi. 2. Documenti archiviati di un'organizzazione. 3. Edificio o istituzione adibiti all'archiviazione di documenti. 4. File che contiene altri file. Cfr. file d'archivio o il sinonimo file contenitore.
Database	<p>Di norma è costituito da uno o più schemi e prevede diritti di accesso a determinate parti del database per utenti o ruoli definiti. In SQL:2008 gli utenti (<i>users</i>) e i ruoli (<i>roles</i>) possono beneficiare di autorizzazioni (privilegi). Un database relazionale è quindi costituito da un insieme di oggetti strutturati (schemi, viste ecc.) e dai contenuti di tabelle.</p> <p>Lo schema del database è una sorta di prefisso dello spazio dei nomi (<i>namespace</i>). Il catalogo del database contiene i metadati di tutti gli schemi nel catalogo. In SQL:2008, il livello catalogo corrisponde al «documento database» che può essere convertito in un formato d'archivio con SIARD.</p>
DATALINK	<p>Un tipo di dati conforme a SQL:2008 parte 9 SQL/MED (ISO/IEC 9075-9:2008). Contiene un riferimento a un LOB in un sistema esterno al database relazionale, ma parzialmente controllato dal sistema di gestione del database relazionale. In SIARD è trattato come un LOB con informazioni sul percorso originale (DLURLPATHONLY).</p>
Dati primari	Dati che costituiscono il contenuto dei documenti. In un file SIARD, la funzione di dati primari è assunta dai dati tabellari.
DIP	Pacchetto di distribuzione (<i>Dissemination Information Package</i>): secondo il modello OAIS, è il contenitore dei dossier ordinati da un utente in una procedura di ordinazione.
DNS	Sistema dei nomi di dominio (<i>Domain Name System</i>): database distribuito che gestisce lo spazio dei nomi (<i>namespace</i>) in Internet.
Documenti	Tutte le informazioni registrate, indipendentemente dal supporto, ricevute o prodotte nell'esecuzione di compiti di servizio pubblico e tutti gli ausili e dati supplementari necessari per la comprensione di tali informazioni e per la loro utilizzazione.
Dossier	Insieme (collettivo) dei documenti relativi a un affare. Di norma, un dossier corrisponde a un affare. Tuttavia, questa struttura di base può essere adeguata a seconda delle esigenze raggruppando affari correlati o suddividendo i dossier in subdossier. La costituzione dei dossier si basa sul sistema di ordinamento.
Fondo	Documenti ripresi dall'archivio per conservazione o archiviati in modo autonomo da altri servizi secondo gli stessi principi.
LOB	Oggetto di grandi dimensioni (<i>large object</i>): termine generico per il contenuto di una cella di una colonna CLOB, BLOB o XML, che può essere rappresentato in un file separato.
MD5	Algoritmo message digest 5 (<i>Message-Digest algorithm 5</i>)
Metadati	Dati che possono essere definiti come «informazioni sui dati primari» (dati sui dati) visto che hanno un carattere descrittivo.

Termine	Descrizione
OAIS	Sistema informativo aperto per l'archiviazione (<i>Open Archival Information System</i>), ISO 14721:2003: modello di riferimento che descrive un archivio come un'organizzazione in cui persone e sistemi interagiscono nel compito di ricevere informazioni e metterle a disposizione di un gruppo di utenti definito.
Pacchetto informativo	Contenitore concettuale costituito da informazioni facoltative sul contenuto e metadati facoltativi sulla sua conservazione. Il pacchetto include informazioni sull'impacchettamento dei dati che separano e identificano le informazioni sul contenuto e la descrizione del pacchetto e che permettono la ricerca di informazioni sul contenuto.
Routine	La routine SQL (nota anche come <i>stored procedure</i>) è importante soprattutto per comprendere le query di visualizzazione, nelle quali può presentarsi sotto forma di espressione parziale.
Schema	Contenitore di tabelle, viste e routine.
SHA1	Algoritmo di hash sicuro (<i>Secure Hash Algorithm</i>)
SIP	Pacchetto di versamento (<i>Submission Information Package</i>): secondo lo standard OAIS, pacchetto informativo trasmesso all'archivio dai servizi produttori. Contiene i documenti digitali (dati primari e secondari).
Soggetto produttore	Servizio o unità organizzativa che ha prodotto e gestito i documenti.
Tabella	Elemento costituito da una definizione con campi che attribuiscono un nome e un tipo a ogni colonna della tabella, da record che contengono i dati tabellari, da una chiave primaria opzionale, da chiavi esterne che assicurano l'integrità referenziale, da chiavi candidate che servono a identificare un record e da vincoli che garantiscono la consistenza. Facoltativamente, per una tabella possono essere definiti cosiddetti trigger.
UTF	Formato di trasformazione Unicode (<i>Unicode Transformation Format</i>)
Vista	Query predefinita memorizzata nel database. Il risultato di una query è una tabella che a sua volta contiene campi e record.
XSD	Definizione dello schema XML (<i>XML Schema Definition</i>)

Allegato C – Documentazione degli standard applicati

eCH-0150	eCH-0150 Change und Release Management von eCH-Standards http://www.ech.ch/
RFC 1738	URL specification – in particular the “file:” URL/URI https://www.ietf.org/rfc/rfc1738.txt
RFC 1951	Specification of the “deflate” algorithm https://www.ietf.org/rfc/rfc1951.txt
RFC 8089	URL specification – The “file” URI Scheme https://tools.ietf.org/html/rfc8089
SQL: 2008	ISO/IEC 9075(1–4,9–11,13,14): 2008: Information technology – Database languages – SQL http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=53681
Unicode	Unicode 6.1.0 Unicode, Inc. http://www.unicode.org/versions/Unicode6.1.0/ (corrisponde a ISO/IEC 10646:2012: Information technology -- Universal Coded Character Set (UCS) http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56921)
XML	Extensible Markup Language (XML), 1.1 (Second Edition) W3C Recommendation 16 August 2006, edited in place 29 September 2006 http://www.w3.org/TR/2006/REC-xml11-20060816/ (corrisponde a ISO/IEC 19503:2005: Information technology -- XML Metadata Interchange (XMI) http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=32622)
ZIP	ZIP File Format Specification, Version 6.3.9 July 15, 2020 PKWARE Inc. http://www.pkware.com/documents/casestudies/APPNOTE.TXT

Allegato D – Estratti dell'esempio ech-0165_oe.siard

Il file SIARD `ech-0165_oe.siard` costituisce un allegato delle specifiche del formato SIARD 2.2 (e 2.1.1). I dati dell'allegato D sono parte di questo file.

D.1 metadata.xsd

La definizione dello schema XML `metadata.xsd` definisce la struttura del file `metadata.xml` nella cartella `header/`.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- =====
XML schema for meta data of the SIARD Format 2.2 RFC
Application: Software-Independent Archival of Relational Databases
Platform : XML 1.0, XML Schema 2001
Description: This XML schema definition defines the structure of the meta data in the SIARD format 2.2.
=====
Copyright : 2007, 2014, 2018, Swiss Federal Archives, Berne, Switzerland, 2020 DILCIS and SFA
===== -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.bar.admin.ch/xmlns/siard/2/metadata.xsd"
targetNamespace="http://www.bar.admin.ch/xmlns/siard/2/metadata.xsd" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.2" id="metadata">
  <!-- 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 address) 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"/>
        <!-- root folder for external files -->
        <xs:element name="lobFolder" type="xs:anyURI" minOccurs="0"/>
        <!-- name and version of program that generated the metadata file -->
        <xs:element name="producerApplication" type="xs:string" minOccurs="0"/>
        <!-- date of creation of archive (automatically generated by SIARD) -->
        <xs:element name="archivalDate" type="xs:date"/>
        <!-- message digest codes over all primary data in folder "content" -->
        <xs:element name="messageDigest" type="messageDigestType" minOccurs="0" maxOccurs="unbounded"/>
        <!-- DNS name of client machine from which connection to the database was established 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 (JDBC URL) 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"/>
        <!-- list of privileges in the archived database -->
        <xs:element name="privileges" type="privilegesType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    <xs:attribute name="version" type="versionType" use="required"/>
    <!-- constraint: version number with release -->
  </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" 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 advanced and structured types in the schema -->
    <xs:element name="types" type="typesType" minOccurs="0"/>
    <!-- list of tables in the schema -->
    <xs:element name="tables" type="tablesType" minOccurs="0"/>
    <!-- list of views in the schema -->
    <xs:element name="views" type="viewsType" minOccurs="0"/>
    <!-- list of routines in the schema -->
    <xs:element name="routines" type="routinesType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type types -->
<xs:complexType name="typesType">
  <xs:annotation>
    <xs:documentation>List of advanced or structured data types types</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="type" type="typeType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type type -->
<xs:complexType name="typeType">
  <xs:annotation>
    <xs:documentation>Advanced or structured data tape type</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- name of data type -->
    <xs:element name="name" type="xs:string"/>
    <!-- category of data type -->
    <xs:element name="category" type="categoryType"/>
    <!-- schema of supertype -->
    <xs:element name="underSchema" type="xs:string" minOccurs="0"/>
    <!-- name of supertype -->
    <xs:element name="underType" type="xs:string" minOccurs="0"/>
    <!-- instantiability if data type (never true for DISTINCT) -->
    <xs:element name="instantiable" type="xs:boolean"/>
    <!-- finality (always true for DISTINCT, never true for structured UDTs) -->
    <xs:element name="final" type="xs:boolean"/>
    <!-- predefined base SQL:2008 type of (DISTINCT) type -->
    <xs:element name="base" type="predefinedTypeType" minOccurs="0"/>
    <!-- alternatively list of attributes (UDT) -->
    <xs:element name="attributes" type="attributesType" 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 attributes -->
<xs:complexType name="attributesType">

```

```

<xs:annotation>
  <xs:documentation>List of attributes of a type</xs:documentation>
</xs:annotation>
<xs:sequence>
  <xs:element name="attribute" type="attributeType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<!-- complex type attribute -->
<xs:complexType name="attributeType">
  <xs:annotation>
    <xs:documentation>Attribute of a type</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the attribute -->
    <xs:element name="name" type="xs:string"/>
    <xs:choice>
      <!-- either predefined or structured -->
      <xs:sequence>
        <!-- SQL:2008 data type of the column -->
        <xs:element name="type" type="predefinedTypeType"/>
      </xs:sequence>
      <xs:sequence>
        <!-- SQL:2008 schema of advanced or structured data type of the attribute -->
        <xs:element name="typeSchema" type="xs:string" minOccurs="0"/>
        <!-- SQL:2008 name of advanced or structured data type of the attribute -->
        <xs:element name="typeName" type="xs:string"/>
      </xs:sequence>
    </xs:choice>
    <!-- original data type of the column -->
    <xs:element name="typeOriginal" type="xs:string" minOccurs="0"/>
    <!-- nullability (default: true) -->
    <xs:element name="nullable" type="xs:boolean" minOccurs="0"/>
    <!-- default value -->
    <xs:element name="defaultValue" type="xs:string" minOccurs="0"/>
    <!-- SQL_1999 cardinality for ARRAY type -->
    <xs:element name="cardinality" type="xs:integer" minOccurs="0"/>
    <!-- description of the attributes's meaning and content -->
    <xs:element name="description" type="xs:string" 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" 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="uniqueKeyType" 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" 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"/>
    <!-- number of rows in the view - added in 2014! -->
    <xs:element name="rows" type="xs:integer" minOccurs="0"/>
  </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" 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"/>
    <!-- folder for LOBs relative to lobFolder of nearest containing
    element for internally or externally stored LOBs -->
    <xs:element name="lobFolder" type="xs:anyURI" minOccurs="0"/>
    <xs:choice>
      <!-- either predefined or structured -->
      <xs:sequence>
        <!-- SQL:2008 predefined data type of the column -->
        <xs:element name="type" type="predefinedTypeType"/>
        <!-- mimeType makes sense only for LOBs and is only informative -->
        <xs:element name="mimeType" type="xs:string" minOccurs="0"/>
      </xs:sequence>
      <xs:sequence>
        <!-- SQL:2008 schema of UDT name of the column -->
        <xs:element name="typeSchema" type="xs:string" minOccurs="0"/>
        <!-- SQL:2008 name of UDT of the column -->
        <xs:element name="typeName" type="xs:string"/>
      </xs:sequence>
    </xs:choice>
    <!-- original data type of the column -->
    <xs:element name="typeOriginal" type="xs:string" minOccurs="0"/>
    <!-- SQL:2008 attribute list of the column -->

```

```

<xs:element name="fields" type="fieldsType" minOccurs="0"/>
<!-- nullability (default: true) -->
<xs:element name="nullable" type="xs:boolean" minOccurs="0"/>
<!-- default value -->
<xs:element name="defaultValue" type="xs:string" minOccurs="0"/>
<!-- SQL_1999 cardinality for ARRAY type -->
<xs:element name="cardinality" type="xs:integer" minOccurs="0"/>
<!-- 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 fields -->
<xs:complexType name="fieldsType">
  <xs:annotation>
    <xs:documentation>List of fields of a column or field</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="field" type="fieldType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type for type of a column or a field -->
<xs:complexType name="fieldType">
  <xs:annotation>
    <xs:documentation>Field element describing the type of a field</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- attribute name or array element position (1-based) -->
    <xs:element name="name" type="xs:string"/>
    <!-- folder for LOBs relative to lobFolder of nearest containing
    element for internally or externally stored LOBs -->
    <xs:element name="lobFolder" type="xs:anyURI" minOccurs="0"/>
    <!-- SQL:2008 sub field list of the field -->
    <xs:element name="fields" type="fieldsType" minOccurs="0"/>
    <!-- mimeType makes sense only for LOBs and is only informative -->
    <xs:element name="mimeType" type="xs:string" minOccurs="0"/>
    <!-- description of the field's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </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" 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" 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="referentialActionType" minOccurs="0"/>
    <!-- ON UPDATE action e.g. ON UPDATE SET DEFAULT -->
    <xs:element name="updateAction" type="referentialActionType" minOccurs="0"/>
    <!-- description of the foreign key's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
  </xs:sequence>

```

```

</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="uniqueKeyType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type uniqueKey -->
<xs:complexType name="uniqueKeyType">
  <xs:annotation>
    <xs:documentation>unique (primary or candidate) key element in siardArchive</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- database name of the unique key -->
    <xs:element name="name" type="xs:string"/>
    <!-- description of the unique key's meaning and content -->
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <!-- columns belonging to the unique key -->
    <xs:element name="column" type="xs:string" 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" maxOccurs="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" 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 BEFORE, AFTER or INSTEAD OF -->
  <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" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type routine -->
<xs:complexType name="routineType">
  <xs:annotation>
    <xs:documentation>Routine</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- specific (unique) name of routine in schema -->
    <xs:element name="specificName" type="xs:string"/>
    <!-- database (possible overloaded) 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:2008 body of routine -->
    <xs:element name="body" type="xs:string" minOccurs="0"/>
    <!-- routine characteristic -->
    <xs:element name="characteristic" type="xs:string" minOccurs="0"/>
    <!-- SQL:2008 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" 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"/>
    <xs:choice>
      <!-- either predefined or structured -->
      <xs:sequence>

```

```

    <!-- SQL:2008 data type of the column -->
    <xs:element name="type" type="predefinedTypeType"/>
  </xs:sequence>
  <xs:sequence>
    <!-- SQL:2008 schema of UDT name of the column -->
    <xs:element name="typeSchema" type="xs:string" minOccurs="0"/>
    <!-- SQL:2008 name of UDT of the column -->
    <xs:element name="typeName" type="xs:string"/>
  </xs:sequence>
</xs:choice>
<!-- original data type of the column -->
<xs:element name="typeOriginal" type="xs:string" minOccurs="0"/>
<!-- SQL_1999 cardinality for ARRAY type -->
<xs:element name="cardinality" type="xs:integer" 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="0" 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" 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" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<!-- complex type privilege -->

```



```

    <xs:enumeration value="SHA-1"/>
    <xs:enumeration value="SHA-256"/>
  </xs:restriction>
</xs:simpleType>
<!-- simple type for version number -->
<xs:simpleType name="versionType">
  <xs:annotation>
    <xs:documentation>versionType is constrained to "2.2" for conformity with this XML schema</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:enumeration value="2.2"/>
    <!-- to be extended later with
  <xs:enumeration value="2.2"/>
  etc. -->
  </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>mandatoryString 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:minLength value="1"/>
    <xs:pattern value="([a-z]|[A-Z])([a-z]|[A-Z]|[0-9]).*/>
  </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="INSTEAD OF"/>
    <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>
<!-- simple type for referential action of a foreign key -->

```

```

<xs:simpleType name="referentialActionType">
  <xs:annotation>
    <xs:documentation>referential action is CASCADE, SET NULL, SET DEFAULT, RESTRICT, or NO ACTION</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="CASCADE"/>
    <xs:enumeration value="SET NULL"/>
    <xs:enumeration value="SET DEFAULT"/>
    <xs:enumeration value="RESTRICT"/>
    <xs:enumeration value="NO ACTION"/>
  </xs:restriction>
</xs:simpleType>
<!-- simple type for the category of a column or a parameter -->
<xs:simpleType name="categoryType">
  <xs:annotation>
    <xs:documentation>category of advanced or structured data types is "distinct" or "udt" for conformity with this XML
schema</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:enumeration value="distinct"/>
    <xs:enumeration value="udt"/>
  </xs:restriction>
</xs:simpleType>
<!-- complex type for the character large object - to be used in table[n].xsd - T_6.2-1 -->
<xs:complexType name="clobType">
  <xs:annotation>
    <xs:documentation source="T_6.2-1" xml:lang="en">a character large object can either be stored inline or as a file internally or externally.
The length is in characters, not in bytes.</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="file" type="xs:anyURI"/>
      <xs:attribute name="length" type="xs:integer"/>
      <xs:attribute name="digestType" type="digestTypeType"/>
      <xs:attribute name="digest" type="xs:string"/>
      <xs:attribute name="dlurlpathonly" type="xs:anyURI"/> <!-- applies only to external LOBs -->
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<!-- complex type for the binary large object - to be used in table[n].xsd -->
<xs:complexType name="blobType">
  <xs:annotation>
    <xs:documentation source="T_6.2-1" xml:lang="en">a binary large object can either be stored inline or as a file internally or externally.
The length is in bytes.</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:hexBinary">
      <xs:attribute name="file" type="xs:anyURI"/>
      <xs:attribute name="length" type="xs:integer"/>
      <xs:attribute name="digestType" type="digestTypeType"/>
      <xs:attribute name="digest" type="xs:string"/>
      <xs:attribute name="dlurlpathonly" type="xs:anyURI"/> <!-- applies only to external LOBs -->
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>

```

D.2 Esempio di metadata.xml

Una possibile descrizione dei metadati per lo schema XML conforme a SIARD di un database si presenta come segue:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<siardArchive xmlns="http://www.bar.admin.ch/xmlns/siard/2/metadata.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.2" xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/2/metadata.xsd metadata.xsd">
  <dbname>OE Sample Database enhanced</dbname>
  <description>Record with PRODUCT_ID 4000 in table PRODUCT_INFORMATION has a picture.version with updated .xsd</description>
  <archiver>Claire Roethlisberger</archiver>
  <archiverContact>claire.roethlisberger@kost.admin.ch</archiverContact>
  <dataOwner>Oracle (OE database) and Swiss Federal Archives (enhancement)</dataOwner>
  <dataOriginTimespan>2000-2007</dataOriginTimespan>
  <producerApplication>SiardGui 2.1.89 Swiss Federal Archives, Berne, Switzerland, 2007-2018</producerApplication>
  <archivalDate>2018-01-30Z</archivalDate>
  <clientMachine>VMW10.entrag.ch</clientMachine>
  <databaseProduct>Oracle Oracle Database 12c Release 12.1.0.1.0 - 64bit Production</databaseProduct>
  <connection>jdbc:oracle:thin:@localhost:1521:ORCL</connection>
  <databaseUser>OE</databaseUser>
  <schemas>
    <schema>
      <name>HR</name>
      <folder>schema0</folder>
      <tables>
        ...
        <table>
          <name>EMPLOYEES</name>
          <folder>table2</folder>
          <description>employees table. Contains 107 rows. References with departments, jobs, job_history tables. Contains a self reference.
          </description>
          <columns>
            <column>
              <name>EMPLOYEE_ID</name>
              <type>DECIMAL(6)</type>
              <typeOriginal>"NUMBER"</typeOriginal>
              <nullable>>false</nullable>
              <description>Primary key of employees table.</description>
            </column>
            <column>
              <name>FIRST_NAME</name>
              <type>VARCHAR(20)</type>
              <typeOriginal>"VARCHAR2"</typeOriginal>
              <description>First name of the employee. A not null column.</description>
            </column>
            <column>
              <name>LAST_NAME</name>
              <type>VARCHAR(25)</type>
              <typeOriginal>"VARCHAR2"</typeOriginal>
              <nullable>>false</nullable>
              <description>Last name of the employee. A not null column.</description>
            </column>
            <column>
              <name>EMAIL</name>
              <type>VARCHAR(25)</type>
              <typeOriginal>"VARCHAR2"</typeOriginal>
              <nullable>>false</nullable>
              <description>Email id of the employee</description>
            </column>
            <column>
              <name>PHONE_NUMBER</name>
              <type>VARCHAR(20)</type>
              <typeOriginal>"VARCHAR2"</typeOriginal>
              <description>Phone number of the employee; includes country code and area code</description>
            </column>
            <column>
              <name>HIRE_DATE</name>
              <type>DATE</type>
              <typeOriginal>"DATE"</typeOriginal>
              <nullable>>false</nullable>
              <description>Date when the employee started on this job. A not null column.</description>
          </columns>
        </table>
      </tables>
    </schema>
  </schemas>
</siardArchive>
```

```

</column>
<column>
  <name>JOB_ID</name>
  <type>VARCHAR(10)</type>
  <typeOriginal>"VARCHAR2"</typeOriginal>
  <nullable>>false</nullable>
  <description>Current job of the employee; foreign key to job_id column of the jobs table. A not null column.</description>
</column>
<column>
  <name>SALARY</name>
  <type>DECIMAL(8, 2)</type>
  <typeOriginal>"NUMBER"</typeOriginal>
  <description>Monthly salary of the employee. Must be greater than zero (enforced by constraint emp_salary_min)</description>
</column>
<column>
  <name>COMMISSION_PCT</name>
  <type>DECIMAL(2, 2)</type>
  <typeOriginal>"NUMBER"</typeOriginal>
  <description>Commission percentage of the employee; Only employees in sales department eligible for commission percentage
  </description>
</column>
<column>
  <name>MANAGER_ID</name>
  <type>DECIMAL(6)</type>
  <typeOriginal>"NUMBER"</typeOriginal>
  <description>Manager id of the employee; has same domain as manager_id in departments table. Foreign key to employee_id
  column of employees table. (useful for reflexive joins and CONNECT BY query)</description>
</column>
<column>
  <name>DEPARTMENT_ID</name>
  <type>DECIMAL(4)</type>
  <typeOriginal>"NUMBER"</typeOriginal>
  <description>Department id where employee works; foreign key to department_id column of the departments table</description>
</column>
</columns>
<primaryKey>
  <name>EMP_EMP_ID_PK</name>
  <column>EMPLOYEE_ID</column>
</primaryKey>
<foreignKeys>
  <foreignKey>
    <name>EMP_DEPT_FK</name>
    <referencedSchema>HR</referencedSchema>
    <referencedTable>DEPARTMENTS</referencedTable>
    <reference>
      <column>DEPARTMENT_ID</column>
      <referenced>DEPARTMENT_ID</referenced>
    </reference>
    <deleteAction>RESTRICT</deleteAction>
    <updateAction>CASCADE</updateAction>
  </foreignKey>
  <foreignKey>
    <name>EMP_MANAGER_FK</name>
    <referencedSchema>HR</referencedSchema>
    <referencedTable>EMPLOYEES</referencedTable>
    <reference>
      <column>MANAGER_ID</column>
      <referenced>EMPLOYEE_ID</referenced>
    </reference>
    <deleteAction>RESTRICT</deleteAction>
    <updateAction>CASCADE</updateAction>
  </foreignKey>
  <foreignKey>
    <name>EMP_JOB_FK</name>
    <referencedSchema>HR</referencedSchema>
    <referencedTable>JOBS</referencedTable>
    <reference>
      <column>JOB_ID</column>
      <referenced>JOB_ID</referenced>
    </reference>
    <deleteAction>RESTRICT</deleteAction>
    <updateAction>CASCADE</updateAction>
  </foreignKey>

```

```

    </foreignKey>
  </foreignKeys>
  <candidateKeys>
    <candidateKey>
      <name>EMP_EMAIL_UK</name>
      <column>EMAIL</column>
    </candidateKey>
  </candidateKeys>
  <rows>107</rows>
</table>
...
</tables>
</schema>
<schema>
  <name>OE</name>
  <folder>schema1</folder>
  <types>
    <type>
      <name>CUST_ADDRESS_TYP</name>
      <category>udt</category>
      <instantiable>true</instantiable>
      <final>true</final>
      <attributes>
        <attribute>
          <name>STREET_ADDRESS</name>
          <type>VARCHAR(40)</type>
        </attribute>
        <attribute>
          <name>POSTAL_CODE</name>
          <type>VARCHAR(10)</type>
        </attribute>
        <attribute>
          <name>CITY</name>
          <type>VARCHAR(30)</type>
        </attribute>
        <attribute>
          <name>STATE_PROVINCE</name>
          <type>VARCHAR(10)</type>
        </attribute>
        <attribute>
          <name>COUNTRY_ID</name>
          <type>CHARACTER(2)</type>
        </attribute>
      </attributes>
    </type>
    <type>
      <name>ORDER_TYP</name>
      <category>udt</category>
      <instantiable>true</instantiable>
      <final>true</final>
      <attributes>
        <attribute>
          <name>ORDER_ID</name>
          <type>DECIMAL(12)</type>
        </attribute>
        <attribute>
          <name>ORDER_MODE</name>
          <type>VARCHAR(8)</type>
        </attribute>
        <attribute>
          <name>CUSTOMER_REF</name>
          <typeSchema>OE</typeSchema>
          <typeName>CUSTOMER_TYP</typeName>
        </attribute>
        <attribute>
          <name>ORDER_STATUS</name>
          <type>DECIMAL(2)</type>
        </attribute>
        <attribute>
          <name>ORDER_TOTAL</name>
          <type>DECIMAL(8, 2)</type>
        </attribute>
      </attributes>
    </type>
  </types>
</schema>

```



```
<attribute>
  <name>SALES_REP_ID</name>
  <type>DECIMAL(6)</type>
</attribute>
<attribute>
  <name>ORDER_ITEM_LIST</name>
  <typeSchema>OE</typeSchema>
  <typeName>ORDER_ITEM_TYP</typeName>
  <cardinality>2147483647</cardinality>
</attribute>
</attributes>
</type>
<type>
  <name>CUSTOMER_TYP</name>
  <category>udt</category>
  <instantiable>true</instantiable>
  <final>true</final>
  <attributes>
    <attribute>
      <name>CUSTOMER_ID</name>
      <type>DECIMAL(6)</type>
    </attribute>
    <attribute>
      <name>CUST_FIRST_NAME</name>
      <type>VARCHAR(20)</type>
    </attribute>
    <attribute>
      <name>CUST_LAST_NAME</name>
      <type>VARCHAR(20)</type>
    </attribute>
    <attribute>
      <name>CUST_ADDRESS</name>
      <typeSchema>OE</typeSchema>
      <typeName>CUST_ADDRESS_TYP</typeName>
    </attribute>
    <attribute>
      <name>PHONE_NUMBERS</name>
      <type>VARCHAR(25)</type>
      <cardinality>5</cardinality>
    </attribute>
    <attribute>
      <name>NLS_LANGUAGE</name>
      <type>VARCHAR(3)</type>
    </attribute>
    <attribute>
      <name>NLS_TERRITORY</name>
      <type>VARCHAR(40)</type>
    </attribute>
    <attribute>
      <name>CREDIT_LIMIT</name>
      <type>DECIMAL(9, 2)</type>
    </attribute>
    <attribute>
      <name>CUST_EMAIL</name>
      <type>VARCHAR(40)</type>
    </attribute>
    <attribute>
      <name>CUST_ORDERS</name>
      <typeSchema>OE</typeSchema>
      <typeName>ORDER_TYP</typeName>
      <cardinality>2147483647</cardinality>
    </attribute>
  </attributes>
</type>
<type>
  <name>ORDER_ITEM_TYP</name>
  <category>udt</category>
  <instantiable>true</instantiable>
  <final>true</final>
  <attributes>
    <attribute>
      <name>ORDER_ID</name>
```

```

    <type>DECIMAL(12)</type>
  </attribute>
  <attribute>
    <name>LINE_ITEM_ID</name>
    <type>DECIMAL(3)</type>
  </attribute>
  <attribute>
    <name>UNIT_PRICE</name>
    <type>DECIMAL(8, 2)</type>
  </attribute>
  <attribute>
    <name>QUANTITY</name>
    <type>DECIMAL(8)</type>
  </attribute>
  <attribute>
    <name>PRODUCT_REF</name>
    <typeSchema>OE</typeSchema>
    <typeName>PRODUCT_INFORMATION_TYP</typeName>
  </attribute>
</attributes>
</type>
<type>
  <name>PRODUCT_INFORMATION_TYP</name>
  <category>udt</category>
  <instantiable>true</instantiable>
  <final>true</final>
  <attributes>
    <attribute>
      <name>PRODUCT_ID</name>
      <type>DECIMAL(6)</type>
    </attribute>
    <attribute>
      <name>PRODUCT_NAME</name>
      <type>VARCHAR(50)</type>
    </attribute>
    <attribute>
      <name>PRODUCT_DESCRIPTION</name>
      <type>VARCHAR(2000)</type>
    </attribute>
    <attribute>
      <name>CATEGORY_ID</name>
      <type>DECIMAL(2)</type>
    </attribute>
    <attribute>
      <name>WEIGHT_CLASS</name>
      <type>DECIMAL(1)</type>
    </attribute>
    <attribute>
      <name>WARRANTY_PERIOD</name>
      <type>INTERVAL YEAR TO MONTH</type>
    </attribute>
    <attribute>
      <name>SUPPLIER_ID</name>
      <type>DECIMAL(6)</type>
    </attribute>
    <attribute>
      <name>PRODUCT_STATUS</name>
      <type>VARCHAR(20)</type>
    </attribute>
    <attribute>
      <name>LIST_PRICE</name>
      <type>DECIMAL(8, 2)</type>
    </attribute>
    <attribute>
      <name>MIN_PRICE</name>
      <type>DECIMAL(8, 2)</type>
    </attribute>
    <attribute>
      <name>CATALOG_URL</name>
      <type>VARCHAR(50)</type>
    </attribute>
  </attributes>

```

```

    <name>INVENTORY_LIST</name>
    <typeSchema>OE</typeSchema>
    <typeName>INVENTORY_TYP</typeName>
    <cardinality>2147483647</cardinality>
  </attribute>
</attributes>
</type>
<type>
  <name>INVENTORY_TYP</name>
  <category>udt</category>
  <instantiable>true</instantiable>
  <final>true</final>
  <attributes>
    <attribute>
      <name>PRODUCT_ID</name>
      <type>DECIMAL(6)</type>
    </attribute>
    <attribute>
      <name>WAREHOUSE</name>
      <typeSchema>OE</typeSchema>
      <typeName>WAREHOUSE_TYP</typeName>
    </attribute>
    <attribute>
      <name>QUANTITY_ON_HAND</name>
      <type>DECIMAL(8)</type>
    </attribute>
  </attributes>
</type>
<type>
  <name>WAREHOUSE_TYP</name>
  <category>udt</category>
  <instantiable>true</instantiable>
  <final>true</final>
  <attributes>
    <attribute>
      <name>WAREHOUSE_ID</name>
      <type>DECIMAL(3)</type>
    </attribute>
    <attribute>
      <name>WAREHOUSE_NAME</name>
      <type>VARCHAR(35)</type>
    </attribute>
    <attribute>
      <name>LOCATION_ID</name>
      <type>DECIMAL(4)</type>
    </attribute>
  </attributes>
</type>
</types>
<tables>
  <table>
    <name>CUSTOMERS</name>
    <folder>table0</folder>
    <description>Contains customers data either entered by an employee or by the customer him/herself over the Web.</description>
    <columns>
      <column>
        <name>CUSTOMER_ID</name>
        <type>DECIMAL(6)</type>
        <typeOriginal>"NUMBER"</typeOriginal>
        <nullable>>false</nullable>
        <description>Primary key column.</description>
      </column>
      <column>
        <name>CUST_FIRST_NAME</name>
        <type>VARCHAR(20)</type>
        <typeOriginal>"VARCHAR2"</typeOriginal>
        <nullable>>false</nullable>
        <description>NOT NULL constraint.</description>
      </column>
      <column>
        <name>CUST_LAST_NAME</name>
        <type>VARCHAR(20)</type>

```

```

<typeOriginal>"VARCHAR2"</typeOriginal>
<nullable>>false</nullable>
<description>NOT NULL constraint.</description>
</column>
<column>
<name>CUST_ADDRESS</name>
<typeSchema>OE</typeSchema>
<typeName>CUST_ADDRESS_TYP</typeName>
<fields>
<field>
<name>STREET_ADDRESS</name>
</field>
<field>
<name>POSTAL_CODE</name>
</field>
<field>
<name>CITY</name>
</field>
<field>
<name>STATE_PROVINCE</name>
</field>
<field>
<name>COUNTRY_ID</name>
</field>
</fields>
<description>Object column of type address_typ.</description>
</column>
<column>
<name>PHONE_NUMBERS</name>
<type>VARCHAR(25)</type>
<typeOriginal>VARCHAR(25) ARRAY[5]</typeOriginal>
<fields>
<field>
<name>PHONE_NUMBERS[1]</name>
</field>
<field>
<name>PHONE_NUMBERS[2]</name>
</field>
<field>
<name>PHONE_NUMBERS[3]</name>
</field>
<field>
<name>PHONE_NUMBERS[4]</name>
</field>
<field>
<name>PHONE_NUMBERS[5]</name>
</field>
</fields>
<cardinality>5</cardinality>
<description>Varray column of type phone_list_typ</description>
</column>
<column>
<name>NLS_LANGUAGE</name>
<type>VARCHAR(3)</type>
<typeOriginal>"VARCHAR2"</typeOriginal>
</column>
<column>
<name>NLS_TERRITORY</name>
<type>VARCHAR(30)</type>
<typeOriginal>"VARCHAR2"</typeOriginal>
</column>
<column>
<name>CREDIT_LIMIT</name>
<type>DECIMAL(9, 2)</type>
<typeOriginal>"NUMBER"</typeOriginal>
<description>Check constraint.</description>
</column>
<column>
<name>CUST_EMAIL</name>
<type>VARCHAR(40)</type>
<typeOriginal>"VARCHAR2"</typeOriginal>
</column>

```

```

<column>
  <name>ACCOUNT_MGR_ID</name>
  <type>DECIMAL(6)</type>
  <typeOriginal>"NUMBER"</typeOriginal>
  <description>References hr.employees.employee_id.</description>
</column>
<column>
  <name>CUST_GEO_LOCATION</name>
  <typeSchema>MDSYS</typeSchema>
  <typeName>SDO_GEOMETRY</typeName>
  <fields>
    <field>
      <name>SDO_GTYPE</name>
    </field>
    <field>
      <name>SDO_SRID</name>
    </field>
    <field>
      <name>SDO_POINT</name>
      <fields>
        <field>
          <name>X</name>
        </field>
        <field>
          <name>Y</name>
        </field>
        <field>
          <name>Z</name>
        </field>
      </fields>
    </field>
    <field>
      <name>SDO_ELEM_INFO</name>
    </field>
    <field>
      <name>SDO_ORDINATES</name>
    </field>
  </fields>
  <description>SDO (spatial) column.</description>
</column>
<column>
  <name>DATE_OF_BIRTH</name>
  <type>DATE</type>
  <typeOriginal>"DATE"</typeOriginal>
</column>
<column>
  <name>MARITAL_STATUS</name>
  <type>VARCHAR(20)</type>
  <typeOriginal>"VARCHAR2"</typeOriginal>
</column>
<column>
  <name>GENDER</name>
  <type>VARCHAR(1)</type>
  <typeOriginal>"VARCHAR2"</typeOriginal>
</column>
<column>
  <name>INCOME_LEVEL</name>
  <type>VARCHAR(20)</type>
  <typeOriginal>"VARCHAR2"</typeOriginal>
</column>
</columns>
<primaryKey>
  <name>CUSTOMERS_PK</name>
  <column>CUSTOMER_ID</column>
</primaryKey>
<foreignKeys>
  <foreignKey>
    <name>CUSTOMERS_ACCOUNT_MANAGER_FK</name>
    <referencedSchema>HR</referencedSchema>
    <referencedTable>EMPLOYEES</referencedTable>
    <reference>
      <column>ACCOUNT_MGR_ID</column>
    </reference>
  </foreignKey>

```

```

        <referenced>EMPLOYEE_ID</referenced>
      </reference>
      <deleteAction>SET NULL</deleteAction>
      <updateAction>CASCADE</updateAction>
    </foreignKey>
  </foreignKeys>
  <rows>319</rows>
</table>
...
<table>
  <name>WAREHOUSES</name>
  <folder>table7</folder>
  <description>Warehouse data unspecific to any industry.</description>
  <columns>
    <column>
      <name>WAREHOUSE_ID</name>
      <type>DECIMAL(3)</type>
      <typeOriginal>"NUMBER"</typeOriginal>
      <nullable>>false</nullable>
      <description>Primary key column.</description>
    </column>
    <column>
      <name>WAREHOUSE_SPEC</name>
      <type>XML</type>
      <typeOriginal>"SYS"."XMLTYPE"</typeOriginal>
    </column>
    <column>
      <name>WAREHOUSE_NAME</name>
      <type>VARCHAR(35)</type>
      <typeOriginal>"VARCHAR2"</typeOriginal>
    </column>
    <column>
      <name>LOCATION_ID</name>
      <type>DECIMAL(4)</type>
      <typeOriginal>"NUMBER"</typeOriginal>
      <description>Primary key column, references hr.locations.location_id.</description>
    </column>
    <column>
      <name>WH_GEO_LOCATION</name>
      <typeSchema>MDSYS</typeSchema>
      <typeName>SDO_GEOMETRY</typeName>
      <fields>
        <field>
          <name>SDO_GTYPE</name>
        </field>
        <field>
          <name>SDO_SRID</name>
        </field>
        <field>
          <name>SDO_POINT</name>
          <fields>
            <field>
              <name>X</name>
            </field>
            <field>
              <name>Y</name>
            </field>
            <field>
              <name>Z</name>
            </field>
          </fields>
        </field>
        <field>
          <name>SDO_ELEM_INFO</name>
        </field>
        <field>
          <name>SDO_ORDINATES</name>
        </field>
      </fields>
      <description>SDO (spatial) column.</description>
    </column>
  </columns>
</table>

```

```

<primaryKey>
  <name>WAREHOUSES_PK</name>
  <column>WAREHOUSE_ID</column>
</primaryKey>
<foreignKeys>
  <foreignKey>
    <name>WAREHOUSES_LOCATION_FK</name>
    <referencedSchema>HR</referencedSchema>
    <referencedTable>LOCATIONS</referencedTable>
    <reference>
      <column>LOCATION_ID</column>
      <referenced>LOCATION_ID</referenced>
    </reference>
    <deleteAction>SET NULL</deleteAction>
    <updateAction>CASCADE</updateAction>
  </foreignKey>
</foreignKeys>
<rows>9</rows>
</table>
</tables>
<views>
  <view>
    <name>ACCOUNT_MANAGERS</name>
    <queryOriginal>SELECT .account_mgr_id _MGR, .region_id, .cust_address.country_id, .cust_address.state_province, (*)
      _CUSTOMERSFROM c, countries crWHERE .cust_address.country_id = cr.country_idGROUP BY ROLLUP (c.account_mgr_id,
      cr.region_id, c.cust_address.country_id, c.cust_address.state_province)</queryOriginal>
    <columns>
      <column>
        <name>ACCT_MGR</name>
        <type>DECIMAL(6)</type>
        <typeOriginal>"NUMBER"</typeOriginal>
      </column>
      <column>
        <name>REGION</name>
        <type>DECIMAL(22)</type>
        <typeOriginal>"NUMBER"</typeOriginal>
      </column>
      <column>
        <name>COUNTRY</name>
        <type>CHARACTER(2)</type>
        <typeOriginal>"CHAR"</typeOriginal>
      </column>
      <column>
        <name>PROVINCE</name>
        <type>VARCHAR(10)</type>
        <typeOriginal>"VARCHAR2"</typeOriginal>
      </column>
      <column>
        <name>NUM_CUSTOMERS</name>
        <type>DECIMAL(22)</type>
        <typeOriginal>"NUMBER"</typeOriginal>
      </column>
    </columns>
    <rows>0</rows>
  </view>
  ...
</views>
<routines>
  <routine>
    <specificName>CATEGORY_DESCRIBE.CATALOG_TYP</specificName>
    <name>CATALOG_TYP</name>
  </routine>
  ...
  <routine>
    <specificName>GET_PHONE_NUMBER_F</specificName>
    <name>GET_PHONE_NUMBER_F</name>
    <returnType>VARCHAR</returnType>
    <parameters>
      <parameter>
        <name>P_IN</name><mode>IN</mode><type>DECIMAL(38)</type><typeOriginal>NUMBER</typeOriginal>
      </parameter>
      <parameter>

```

```
<name>P_PHONELIST</name><mode>IN</mode><type>VARCHAR(25)</type><cardinality>5</cardinality>
</parameter>
</parameters>
</routine>
...
</routines>
</schema>
<schema>
...
</schema>
</schemas>
<users>
<user><name>OE</name></user>
<user><name>HR</name></user>
</users>
<roles>
<role><name>BI</name><admin>OE</admin></role>
<role><name>PM</name><admin>OE</admin></role>
</roles>
<privileges>
<privilege>
<type>REFERENCES</type>
<object>HR.COUNTRIES</object>
<grantor>HR</grantor>
<grantee>OE</grantee>
</privilege>
...
</privileges>
</siardArchive>
```


D.3 Esempi di definizione dello schema XML di una tabella

Per ogni tabella, SIARD genera una definizione dello schema XML che attribuisce i tipi di dati XML corretti alle colonne.

D.3a table2.xsd (definizione dello schema di una tabella semplice)

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<xs:schema xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.2">
  <!-- root element is the table element -->
  <xs:element name="table">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>Root element of a table of the SIARD archive. A table consists of rows.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="row" type="recordType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="versionType" use="required"/>
    </xs:complexType>
  </xs:element>
  <!-- simple type for version number -->
  <xs:simpleType name="versionType">
    <xs:annotation>
      <xs:documentation>versionType is constrained to "2.2" for conformity with this XML schema</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
      <xs:enumeration value="2.2"/>
      <!-- to be extended later with <xs:enumeration value="2.2"/> etc. -->
    </xs:restriction>
  </xs:simpleType>
  <!-- complex type record -->
  <xs:complexType name="recordType">
    <xs:annotation>
      <xs:documentation>row type of a table of the SIARD archive. A row consists of columns.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="c1" type="xs:decimal"/>
      <xs:element name="c2" type="xs:string" minOccurs="0"/>
      <xs:element name="c3" type="xs:string"/>
      <xs:element name="c4" type="xs:string"/>
      <xs:element name="c5" type="xs:string" minOccurs="0"/>
      <xs:element name="c6" type="dateType"/>
      <xs:element name="c7" type="xs:string"/>
      <xs:element name="c8" type="xs:decimal" minOccurs="0"/>
      <xs:element name="c9" type="xs:decimal" minOccurs="0"/>
      <xs:element name="c10" type="xs:decimal" minOccurs="0"/>
      <xs:element name="c11" type="xs:decimal" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!-- date type between 0001 and 9999 restricted to UTC -->
  <xs:simpleType name="dateType">
    <xs:restriction base="xs:date">
      <xs:minInclusive value="0001-01-01Z"/>
      <xs:maxExclusive value="10000-01-01Z"/>
      <xs:pattern value="\d{4}-\d{2}-\d{2}Z?"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

D.3b table7.xsd (definizione dello schema di una tabella con *large object* interni)

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<xs:schema xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.2">
  <!-- root element is the table element -->
  <xs:element name="table">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>Root element of a table of the SIARD archive. A table consists of rows.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="row" type="recordType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="versionType" use="required"/>
    </xs:complexType>
  </xs:element>
  <!-- simple type for version number -->
  <xs:simpleType name="versionType">
    <xs:annotation>
      <xs:documentation>versionType is constrained to "2.2" for conformity with this XML schema</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
      <xs:enumeration value="2.2"/>
      <!-- to be extended later with <xs:enumeration value="2.2"/> etc. -->
    </xs:restriction>
  </xs:simpleType>
  <!-- complex type record -->
  <xs:complexType name="recordType">
    <xs:annotation>
      <xs:documentation>row type of a table of the SIARD archive. A row consists of columns.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="c1" type="xs:decimal"/>
      <xs:element name="c2" type="clobType" minOccurs="0"/>
      <xs:element name="c3" type="xs:string" minOccurs="0"/>
      <xs:element name="c4" type="xs:decimal" minOccurs="0"/>
      <xs:element name="c5" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="u1" type="xs:decimal" minOccurs="0"/>
            <xs:element name="u2" type="xs:decimal" minOccurs="0"/>
            <xs:element name="u3" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="u1" type="xs:decimal" minOccurs="0"/>
                  <xs:element name="u2" type="xs:decimal" minOccurs="0"/>
                  <xs:element name="u3" type="xs:decimal" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="u4" type="xs:decimal" minOccurs="0"/>
            <xs:element name="u5" type="xs:decimal" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <!-- type for text large objects -->
  <xs:complexType name="clobType">
    <xs:annotation>
      <xs:documentation>a text large object can either be stored inline (as xs:string) or externally (addressed by URI). The digest makes sure,
that the connection to the external object is not completely lost. The length is in characters, not in bytes.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="file" type="xs:anyURI"/>
        <xs:attribute name="length" type="xs:integer"/>
        <xs:attribute name="digestType" type="digestType"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

```

```

    <xs:attribute name="digest" type="xs:string"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<!-- type for message digest type -->
<xs:simpleType name="digestTypeType">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:enumeration value="MD5"/>
    <xs:enumeration value="SHA-1"/>
    <xs:enumeration value="SHA-256"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

D.3c table0.xsd (definizione dello schema di una tabella con *udt* e array)

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<xs:schema xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.2">
  <!-- root element is the table element -->
  <xs:element name="table">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>Root element of a table of the SIARD archive. A table consists of rows.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="row" type="recordType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="versionType" use="required"/>
    </xs:complexType>
  </xs:element>
  <!-- simple type for version number -->
  <xs:simpleType name="versionType">
    <xs:annotation>
      <xs:documentation>versionType is constrained to "2.2" for conformity with this XML schema</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
      <xs:enumeration value="2.2"/>
      <!-- to be extended later with <xs:enumeration value="2.2"/> etc. -->
    </xs:restriction>
  </xs:simpleType>
  <!-- complex type record -->
  <xs:complexType name="recordType">
    <xs:annotation>
      <xs:documentation>row type of a table of the SIARD archive. A row consists of columns.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="c1" type="xs:decimal"/>
      <xs:element name="c2" type="xs:string"/>
      <xs:element name="c3" type="xs:string"/>
      <xs:element name="c4" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="u1" type="xs:string" minOccurs="0"/>
            <xs:element name="u2" type="xs:string" minOccurs="0"/>
            <xs:element name="u3" type="xs:string" minOccurs="0"/>
            <xs:element name="u4" type="xs:string" minOccurs="0"/>
            <xs:element name="u5" type="xs:string" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="c5" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="a1" type="xs:string" minOccurs="0"/>
            <xs:element name="a2" type="xs:string" minOccurs="0"/>
            <xs:element name="a3" type="xs:string" minOccurs="0"/>
            <xs:element name="a4" type="xs:string" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```
<xs:element name="a5" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="c6" type="xs:string" minOccurs="0"/>
<xs:element name="c7" type="xs:string" minOccurs="0"/>
<xs:element name="c8" type="xs:decimal" minOccurs="0"/>
<xs:element name="c9" type="xs:string" minOccurs="0"/>
<xs:element name="c10" type="xs:decimal" minOccurs="0"/>
<xs:element name="c11" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="u1" type="xs:decimal" minOccurs="0"/>
      <xs:element name="u2" type="xs:decimal" minOccurs="0"/>
      <xs:element name="u3" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="u1" type="xs:decimal" minOccurs="0"/>
            <xs:element name="u2" type="xs:decimal" minOccurs="0"/>
            <xs:element name="u3" type="xs:decimal" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="u4" type="xs:decimal" minOccurs="0"/>
      <xs:element name="u5" type="xs:decimal" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="c12" type="dateType" minOccurs="0"/>
<xs:element name="c13" type="xs:string" minOccurs="0"/>
<xs:element name="c14" type="xs:string" minOccurs="0"/>
<xs:element name="c15" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<!-- date type between 0001 and 9999 restricted to UTC -->
<xs:simpleType name="dateType">
  <xs:restriction base="xs:date">
    <xs:minInclusive value="0001-01-01Z"/>
    <xs:maxExclusive value="10000-01-01Z"/>
    <xs:pattern value="\d{4}-\d{2}-\d{2}Z?"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

D.4 Esempi di dati tabellari di una tabella

I dati tabellari sono memorizzati in un file XML che soddisfa la definizione dello schema XML della tabella.

D.4a table2.xml (tabella semplice)

```
<?xml version="1.0" encoding="UTF-8"?>
<table xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/2/table.xsd table2.xsd" version="2.2">
  <row><c1>100</c1><c2>Steven</c2><c3>King</c3><c4>SKING</c4><c5>515.123.4567</c5>
    <c6>2003-06-16Z</c6><c7>AD_PRES</c7><c8>24000</c8><c10>100</c10><c11>90</c11></row>
  <row><c1>101</c1><c2>Neena</c2><c3>Kochhar</c3><c4>NKOCHHAR</c4><c5>515.123.4568</c5>
    <c6>2005-09-20Z</c6><c7>AD_VP</c7><c8>17000</c8><c10>100</c10><c11>90</c11></row>
  <row><c1>102</c1><c2>Lex</c2><c3>De Haan</c3><c4>LDEHAAN</c4><c5>515.123.4569</c5>
    <c6>2001-01-12Z</c6><c7>AD_VP</c7><c8>17000</c8><c10>100</c10><c11>90</c11></row>
  <row><c1>103</c1><c2>Alexander</c2><c3>Hunold</c3><c4>AHUNOLD</c4><c5>590.423.4567</c5>
    <c6>2006-01-02Z</c6><c7>IT_PROG</c7><c8>9000</c8><c10>102</c10><c11>60</c11></row>
  <row><c1>104</c1><c2>Bruce</c2><c3>Ernst</c3><c4>BERNST</c4><c5>590.423.4568</c5>
    <c6>2007-05-20Z</c6><c7>IT_PROG</c7><c8>6000</c8><c10>103</c10><c11>60</c11></row>
  <row><c1>105</c1><c2>David</c2><c3>Austin</c3><c4>DAUSTIN</c4><c5>590.423.4569</c5>
    <c6>2005-06-24Z</c6><c7>IT_PROG</c7><c8>4800</c8><c10>103</c10><c11>60</c11></row>
  <row><c1>106</c1><c2>Valli</c2><c3>Pataballa</c3><c4>VPATABAL</c4><c5>590.423.4560</c5>
    <c6>2006-02-04Z</c6><c7>IT_PROG</c7><c8>4800</c8><c10>103</c10><c11>60</c11></row>
  <row><c1>107</c1><c2>Diana</c2><c3>Lorentz</c3><c4>DLORENTZ</c4><c5>590.423.5567</c5>
    <c6>2007-02-06Z</c6><c7>IT_PROG</c7><c8>4200</c8><c10>103</c10><c11>60</c11></row>
  <row><c1>108</c1><c2>Nancy</c2><c3>Greenberg</c3><c4>NGREENBE</c4><c5>515.124.4569</c5>
    <c6>2002-08-16Z</c6><c7>FI_MGR</c7><c8>12008</c8><c10>101</c10><c11>100</c11></row>
  ...
  <row><c1>206</c1><c2>William</c2><c3>Gietz</c3><c4>WGIETZ</c4><c5>515.123.8181</c5><c6>2002-06-
06Z</c6><c7>AC_ACCOUNT</c7><c8>8300</c8><c10>205</c10><c11>110</c11></row>
</table>
```

D.4b table7.xml (tabella con *large object* interni)

```
<?xml version="1.0" encoding="UTF-8"?>
<table xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/2/table.xsd table7.xsd" version="2.2">
  <row>
    <c1>1</c1>
    <c2 file="content/schema1/table7/lob1/record0.xml" length="270" digestType="MD5" digest="BCA4FB6D6898A2F42C624839B431C386"/>
    <c3>Southlake, Texas</c3>
    <c4>1400</c4>
    <c5><u1>2001</u1><u2>8307</u2><u3><u1>-103.00195</u1><u2>36.500374</u2></u3></c5>
  </row>
  <row>
    <c1>2</c1>
    <c2 file="content/schema1/table7/lob1/record1.xml" length="268" digestType="MD5" digest="7E99F05D8C4D7D3909D3F20987A0DE41"/>
    <c3>San Francisco</c3>
    <c4>1500</c4>
    <c5><u1>2001</u1><u2>8307</u2><u3><u1>-124.21014</u1><u2>41.998016</u2></u3></c5>
  </row>
  <row>
    <c1>3</c1>
    <c2 file="content/schema1/table7/lob1/record2.xml" length="235" digestType="MD5" digest="C495BB25A6EDBFE829DDB9B28C027DC3"/>
    <c3>New Jersey</c3>
    <c4>1600</c4>
    <c5><u1>2001</u1><u2>8307</u2><u3><u1>-74.695305</u1><u2>41.35733</u2></u3></c5>
  </row>
  ...
  <row>
    <c1>9</c1><c3>Bombay</c3><c4>2100</c4>
  </row>
</table>
```

D.4c table0.xml (tabella con *udt* e array)

```

<?xml version="1.0" encoding="UTF-8"?>
<table xmlns="http://www.bar.admin.ch/xmlns/siard/2/table.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/2/table.xsd table0.xsd" version="2.2">
  <row>
    <c1>232</c1>
    <c2>Donald</c2>
    <c3>Hunter</c3>
    <c4><u1>5122 Sinclair Ln</u1><u2>21206</u2><u3>Baltimore</u3><u4>MD</u4><u5>US</u5></c4>
    <c5><a1>+1 410 123 4795</a1></c5>
    <c6>us</c6>
    <c7>AMERICA</c7>
    <c8>2400</c8>
    <c9>Donald.Hunter@CHACHALACA.EXAMPLE.COM</c9>
    <c10>145</c10>
    <c11><u1>2001</u1><u2>8307</u2><u3><u1>-76.545732</u1><u2>39.322775</u2></u3></c11>
    <c12>1960-01-19Z</c12>
    <c13>married</c13>
    <c14>M</c14>
    <c15>G: 130,000 - 149,999</c15>
  </row>
  <row>
    <c1>233</c1>
    <c2>Graham</c2>
    <c3>Spielberg</c3>
    <c4><u1>680 Bel Air Rd</u1><u2>21014</u2><u3>Bel Air</u3><u4>MD</u4><u5>US</u5></c4>
    <c5><a1>+1 410 123 4800</a1></c5>
    <c6>us</c6>
    <c7>AMERICA</c7>
    <c8>2400</c8>
    <c9>Graham.Spielberg@CHUKAR.EXAMPLE.COM</c9>
    <c10>145</c10>
    <c11><u1>2001</u1><u2>8307</u2><u3><u1>-76.357073</u1><u2>39.523878</u2></u3></c11>
    <c12>1970-01-28Z</c12>
    <c13>single</c13>
    <c14>M</c14>
    <c15>D: 70,000 - 89,999</c15>
  </row>
  ...
  <row>
    <row>
      <c1>235</c1>
      <c2>Edward</c2>
      <c3>Oates</c3>
      <c4><u1>8004 Stansbury Rd</u1><u2>21222</u2><u3>Baltimore</u3><u4>MD</u4><u5>US</u5></c4>
      <c5><a1>+1 410 012 4715</a1><a2>+1 410 083 4715</a2></c5>
      <c6>us</c6>
      <c7>AMERICA</c7>
      <c8>2400</c8>
      <c9>Edward.Oates@OVENBIRD.EXAMPLE.COM</c9>
      <c10>145</c10>
      <c11><u1>2001</u1><u2>8307</u2><u3><u1>-76.500344</u1><u2>39.25618</u2></u3></c11>
      <c12>1955-03-20Z</c12>
      <c13>married</c13>
      <c14>M</c14>
      <c15>E: 90,000 - 109,999</c15>
    </row>
    ...
  </table>

```

Allegato E – Esempio di segmentazione di LOB interni

Di seguito è riportato un esempio di struttura delle cartelle per il database *Northwind* con LOB interni memorizzati al di fuori del file SIARD. *Northwind* include le tabelle seguenti, ordinate in questo modo nel presente esempio:

Orders	(table0)
Products	(table1)
Categories	(table2) – con LOB con più di 2000 byte o caratteri
Shippers	(table3)
Employees	(table4) – con LOB con più di 2000 byte o caratteri
Territories	(table5)
CustomerDemographics	(table6)
CustomerCustomerDemo	(table7)
Suppliers	(table8)
EmployeeTerritories	(table9)
Customers	(table10)
Sysdiagrams	(table11)
Region	(table12)

Nel presente esempio è impiegata soltanto la tabella *Categories* (table2)

CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	BLOB (Size: 10151)
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	BLOB (Size: 12107)
3	Confections	Desserts, candies, and sweet breads	BLOB (Size: 12007)
4	Dairy Products	Cheeses	BLOB (Size: 9756)
5	Grains/Cereals	Breads, crackers, pasta, and cereal	BLOB (Size: 12131)
6	Meat/Poultry	Prepared meats	BLOB (Size: 11280)
7	Produce	Dried fruit and bean curd	BLOB (Size: 12338)
8	Seafood	Seaweed and fish	BLOB (Size: 12069)

Limitazione del numero e delle dimensioni totali dei file per cartella

Nel presente esempio il numero massimo di file per cartella è fissato a 4 e le dimensioni totali massime dei file per cartella a 45 000 byte (non realistico, ma utile come esempio).

I LOB della colonna 4 (*Picture*) delle righe 1, 2, 3 e 4 (t2_c4_r5.bin, t2_c4_r6.bin, t2_c4_r7.bin) sono memorizzati insieme in una cartella denominata seg_0

Dato che il *numero massimo di 4 file* è superato, non possono essere memorizzati altri file in questa cartella e viene quindi creata una nuova cartella seg_1.

I LOB della colonna 4 (*Picture*) delle righe 5, 6 e 7 (t2_c4_r5.bin, t2_c4_r6.bin, t2_c4_r7.bin) sono memorizzati insieme nella cartella seg_1.

Il LOB della colonna 4 (*Picture*) della riga 8 (t2_c4_r8.bin) non è memorizzato insieme ai LOB delle righe 5, 6 e 7 nella cartella seg_1. Questo non perché è stato superato il *numero massimo di 4 file*, ma perché sono superate le *dimensioni massime dei file per cartella* di 45 000 byte.

Le dimensioni dei LOB delle righe 5, 6 e 7 sono rispettivamente di 12,131, 11 280 e 12 338 byte per un totale di 35 749 byte. Se si aggiunge il LOB della riga 8, le cui dimensioni sono di 12 069 byte, ai 35 749 byte delle righe 5, 6 e 7 si arriva a un totale di 47 818 byte e si superano quindi le *dimensioni massime per cartella* di 45 000 byte. Di conseguenza è creata una nuova cartella `seg_2`, in cui è memorizzato il LOB della colonna 4 (*Picture*) della riga 8 (`t2_c4_r8.bin`).

Illustrazione della struttura delle cartelle dell'esempio

```
Northwind.siard <!-- packaged as a ZIP file -->
  content/
  header/
    metadata.xml
    metadata.xsd
    siardversion/
      2.2/

Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r1.bin
Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r2.bin
Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r3.bin
Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r4.bin <!-- folder file number
limit -->
Northwind_lobs/s0_t2_c4/seg_1/t2_c4_r5.bin
Northwind_lobs/s0_t2_c4/seg_1/t2_c4_r6.bin
Northwind_lobs/s0_t2_c4/seg_1/t2_c4_r7.bin <!-- folder file size
limit -->
Northwind_lobs/s0_t2_c4/seg_2/t2_c4_r8.bin
```

Estratto del file metadata.xml dell'esempio

```
<siardArchive>...<lobFolder>./Northwind_lobs/</lobFolder>
<dbname>Northwind</dbname>

..
<column>...<lobFolder>s0_t2_c4/</lobFolder>...</column>
```

Estratto del file table2.xml dell'esempio

```
<row><c1>1</c1><c2>Beverages</c2><c3>Soft drinks, coffees, teas, beers, and ales</c3>
<c4 file="seg_0/t2_c4_r1.bin" length="10151" digestType="md5"
digest="74f24080fc9d234d3ac221b8e743c763"/></row>
<row><c1>2</c1><c2>Condiments</c2><c3>Sweet and savory sauces, relishes, spreads, and
seasonings</c3>
<c4 file="seg_0/t2_c4_r2.bin" length="12107" digestType="md5"
digest="22a0cbe8960b78ce48b07a285ce69e3c"/></row>
<row><c1>3</c1><c2>Confections</c2><c3>Desserts, candies, and sweet breads</c3>
<c4 file="seg_0/t2_c4_r3.bin" length="12007" digestType="md5"
digest="3e2f2028a9147c29bdcd36ed4e5f25b3"/></row>
<row><c1>4</c1><c2>Dairy Products</c2><c3>Cheeses</c3>
<c4 file="seg_0/t2_c4_r4.bin" length="9756" digestType="md5"
digest="12f588040e11cc2021ea37d46aa10c51"/></row>
```



```

<row><c1>5</c1><c2>Grains/Cereals</c2><c3>Breads, crackers, pasta, and cereal</c3>
<c4 file="seg_1/t2_c4_r5.bin" length="12131" digestType="md5"
digest="e2d8ef03e1b24edd946820dbbf44dfd"/></row>
<row><c1>6</c1><c2>Meat/Poultry</c2><c3>Prepared meats</c3>
<c4 file="seg_1/t2_c4_r6.bin" length="11280" digestType="md5"
digest="814a3eb95253c08137f70bcfc279e00f"/></row>
<row><c1>7</c1><c2>Produce</c2><c3>Dried fruit and bean curd</c3>
<c4 file="seg_1/t2_c4_r7.bin" length="12338" digestType="md5"
digest="ee114cd7700f566b1f7c7e8e0f68ca0f"/></row>
<row><c1>8</c1><c2>Seafood</c2><c3>Seaweed and fish</c3>
<c4 file="seg_2/t2_c4_r8.bin" length="12069" digestType="md5"
digest="2de1ac4c4e8ebb853e17db01af3fb7c3"/></row>

```

Risoluzione dell'URL dell'esempio

Nel formato SIARD, l'URI di base e il riferimento del percorso della riga 4 dell'esempio è risolto come segue in conformità a RFC 8089:

Base URI: ./Northwind_lobs/

Column URI: s0_t2_c4/

File URI: seg_0/t2_c4_r4.bin

Resolves into: ./Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r4.bin

dove il ./ dell'URI di base ./Northwind_lobs/ corrisponde alla collocazione del file metadata.xml nella cartella header all'interno della cartella ZIP Northwind.siard; la cartella Northwind_lobs/ si trova quindi allo stesso livello del file Northwind.siard.

Prescrizioni dello standard XML per gli spazi di escape

Il formato SIARD 2.2 usa il tipo di dati XML *xs:anyURI* per rappresentare gli URI. Secondo la raccomandazione del W3C questo tipo di dati può contenere valori, che possono essere assoluti o relativi, e, a titolo facoltativo, identificatori di frammenti. La raccomandazione del W3C rinvia a RFC 2396²¹ e RFC 2732²² per le definizioni. Questi due standard si situano tra le RFC 1738 e le più recenti RFC 8089, già presentate in questa raccomandazione.

In base a una nota inclusa nella raccomandazione del W3C, lo spazio lessicale, ossia l'insieme dei letterali validi per il tipo di dati, ammette di norma gli spazi, tuttavia il loro uso è altamente sconsigliato a meno che non siano codificati come %20. Per gli spazi negli URI il formato SIARD 2.2 segue la raccomandazione del W3C, ma è raccomandato di non inserire caratteri di escape nel contenuto degli elementi *lobFolder*, come mostrato nel seguente esempio: Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r4.bin.

²¹ <https://www.w3.org/TR/xmlschema-2/#RFC2396>

²² <https://www.w3.org/TR/xmlschema-2/#RFC2732>

Algoritmo preposto al valore di hashing richiesto in SIARD-2.0 e modificato in SIARD-2.1

In base alle specifiche del formato SIARD-2.0, al valore del codice *message digest* deve essere preposto l'algoritmo seguito dal valore di hashing. Dalla versione SIARD 2.1 il valore e l'algoritmo sono separati.

Lettere minuscole raccomandate per l'attributo *messageDigest*

Il codice *message digest* indica valori esadecimali e quindi non è di stretta rilevanza che sia scritto in lettere maiuscole o minuscole. Sono tuttavia utilizzate e raccomandate per lo più le lettere minuscole, si veda per esempio RFC 2831:

<https://www.ietf.org/rfc/rfc2831.txt>.

Allegato F – Gestione di *large object* in database relazionali

Large object (LOB) è il nome comunemente usato per i *binary large object* (BLOB) e i *character large object* (CLOB). I BLOB sono contenuti di tipo video, suoni, immagini, documenti di elaborazione testi ecc. e i CLOB di tipo testuale. I LOB possono essere memorizzati all'interno, sotto forma di BLOB o CLOB, o all'esterno, sotto forma di DATALINK, di un database relazionale (SQL:2008 SQL/MED).

Nel contesto dei database relazionali, i dati binari sono definiti come dati per i quali non esiste alcun tipo di dati semplice (p. es. numeri interi o caratteri). Sono inoltre rilevanti anche le loro dimensioni a causa della necessità di una loro gestione efficace nei database. I dati binari sono generalmente denominati BLOB.

In analogia, grandi quantità di dati formati da caratteri sono chiamati CLOB. A porre problemi sono soprattutto le loro dimensioni, più che la mancanza di un tipo di dati appropriato. Nelle presenti specifiche i CLOB sono trattati come BLOB, ed entrambi sono denominati LOB.

I database e la loro gestione dei LOB sono sempre stati una sfida, indipendentemente dal fatto che la gestione sia basata su:

1. LOB interni - contenuti nei record del database relazionale;
2. LOB esterni - memorizzati in file al di fuori del database relazionale e referenziati mediante un percorso (URL).

Gestione di LOB secondo gli standard SQL

Il primo metodo, che usa LOB interni, è disponibile da tempo nelle versioni successive dello standard SQL. È supportato da tutti gli attuali sistemi di gestione di database relazionali.

Un secondo metodo, che usa file esterni, è disponibile da SQL:2003 ed è denominato *Management of External Data* (SQL/MED). È parzialmente supportato dagli attuali sistemi di gestione di database relazionali; i sistemi che lo supportano parzialmente, lo fanno in maniera diversa, probabilmente perché lo standard SQL non fornisce dettagli.

Gestione di LOB nel formato SIARD-2.2

Il formato SIARD-2.2 verte sullo standard SQL:2008.

LOB interni (ISO/IEC 9075-2:2008 - BLOBS) supportati in SIARD-2.2

Come era già il caso per la versione 1.0 (SQL:1999), le specifiche del formato SIARD-2.2 supportano il metodo SQL:2008 per l'impiego di LOB interni (ISO/IEC 9075-2:2008).

Il formato SIARD-2.2 supporta i LOB memorizzati sotto forma di file all'interno del file SIARD, come descritto in dettaglio nelle specifiche del formato SIARD-2.2 (in analogia a SIARD-1.0).

Il formato SIARD-2.2 supporta i LOB memorizzati sotto forma di file al fuori del file SIARD (funzionalità introdotta in SIARD-2.0), come descritto in dettaglio nelle presenti specifiche.

File esterni (ISO/IEC 9075-9:2008 – SQL/MED) supportati in SIARD-2.2

Il formato SIARD-2.2 supporta il metodo SQL:2008 per l'impiego di file esterni (ISO/IEC 9075-9:2008 - SQL/MED).

Allegato G – SIARD e ZIP

La specifica SIARD G_3.2-1 stabilisce che «*un database relazionale è archiviato in un unico file SIARD*». Questo, tuttavia, non significa necessariamente che il file non possa essere segmentato. SIARD usa il formato ZIP (32 bit e 64 bit) che supporta la suddivisione di un file in segmenti ZIP con determinate dimensioni (max. 4 GiB per segmento per ZIP64)²³.

Il formato ZIP64 dovrebbe pertanto essere adatto alla suddivisione di un file SIARD molto voluminoso in segmenti gestibili:

```
myDatabase.siard.z1  
myDatabase.siard.z2  
...  
myDatabase.siard.z (n-1)  
myDatabase.siard.zip
```

L'ultimo segmento del file ZIP contiene il repertorio dell'archivio ZIP. I file ZIP possono anche essere trasmessi in streaming.

Considerate le funzionalità del formato ZIP, per suddividere un file zip in segmenti, unitamente alla capacità del formato ZIP64 di supportare file di grandi dimensioni, al repertorio centrale e alla possibilità di trasmissione in streaming, il formato ZIP64 dovrebbe costituire una soluzione sufficientemente adeguata per gestire grandi file SIARD.

Si spera che in un futuro molto prossimo sarà sufficiente utilizzare queste funzionalità integrate, ma attualmente non è ancora possibile, tra l'altro per i seguenti motivi:

La corretta suddivisione di un archivio ZIP in segmenti in conformità alle specifiche ZIP64 non è ancora generalmente supportata dalle applicazioni (inclusi gli strumenti di programmazione), per esempio limitazione del numero massimo di segmenti o delle dimensioni massime del segmento (che a volte sono molto inferiori ai valori massimi previsti dalle specifiche del formato – attualmente soltanto 4 GiB, che è un altro problema).

ISO/IEC 21320-1:2015(E) *Document Container File* supporta ZIP64, ma non la sezione 8.0 di Appnote (*Splitting and spanning Zip files*).

In generale, le applicazioni non supportano ancora una gestione efficace dei segmenti, per esempio è necessario creare un file ZIP completo prima di poterlo dividere o assemblare l'intero file zip prima poter procedere all'indirizzamento o all'estrazione di sue parti.

Non appena le applicazioni supporteranno determinate funzionalità in misura ritenuta sufficiente, sarà pubblicata una nuova versione delle presenti specifiche basata sulla segmentazione automatica; nel frattempo continua ad applicarsi la segmentazione manuale come specificato nel presente documento.

²³ https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT_section_8.5.1

Numero massimo di segmenti = 4 294 967 295 - 1; dimensioni massime dei segmenti ZIP = 4 294 967 295 byte

Allegato H – Modifiche rispetto alla versione 1.0

Alla versione 2.2 sono state apportate le seguenti modifiche rispetto alla versione 1.0²⁴.

Capitolo / ID / documento	Adeguamento	RFC
Passim	Requisiti facoltativi: viene chiaramente indicato se un campo è obbligatorio o meno e se può anche essere lasciato vuoto.	2013-23
Passim, cap. 5.3, 5.4, 5.7, 6.2 e 6.3	Il passaggio da SQL:1999 a SQL:2008 ha un'influenza su quasi tutto e richiede anche nuovi capitoli.	2014-110
Frontespizio, cap. 3-6, allegato D	Tutti gli esempi si riferiscono al nuovo esempio ech-0165_oe.siard allegato.	
Sintesi	Aggiornata e corredata di spiegazioni sull'uso delle versioni precedenti	
Cap. 1 e 2	Numerazione dei capitoli in base al nuovo modello	
Cap. 2.2, passim	ID del requisito G al posto di A	
G_3.2-1, G_3.4-3, P_4.2-3, passim, metadata.xsd	Adeguamenti nella sezione relativa ai <i>large object</i> , inclusi gli attributi <i>digest</i> e <i>digestType</i> . Con la nuova versione è possibile memorizzare esternamente, ossia al fuori del file SIARD, i BLOB e i LOB.	2015-29
G_3.2-2	Soppresso poiché non si tratta di una definizione di formato, ma di un requisito organizzativo.	
G_4.1-2	Compressione deflate consentita e raccomandata	Aggiunta
P_4.2-4, P_4.2-5	Riconoscimento del formato: per facilitare il riconoscimento del formato SIARD (p. es. mediante PRONOM), il file deve contenere una cartella vuota <code>header/siardversion/2.1/</code> che indica la versione del formato SIARD.	2015-12
P_4.3-3, passim, metadata.xsd	Passaggio da SQL:1999 a SQL:2008. Sono state integrate le abbreviazioni SQL:2008.	2014-110
Cap. 5.4, metadata.xsd	Inserito elemento nullable dell'attributo	
Cap. 5.6 e 5.7	Nella sezione relativa alle colonne e ai campi, <i>folder</i> è stato sostituito con <i>lobFolder</i> .	
T_6.1-2, T_6.1-4	Precisazione sull'inizio della numerazione progressiva	
T_6.1-3, T_6.3-2, passim, metadata.xsd	Adeguamenti nella sezione «Date, orari e marche temporali».	
Cap. 5.13, metadata.xsd	Ampliata l'actionTime del trigger con INSTEAD OF	

²⁴ Le modifiche tra le versioni 2.1 e 2.1.1 si limitano a precisazioni nella formulazione della sintesi e in G_3.3-4.

Capitolo / ID / documento	Adeguamento	RFC
metadata.xsd	<p>La cartella metadata.xsd è stata adeguata in diversi punti, tra l'altro per i motivi seguenti: adeguamento allo standard SQL:2008 (designazioni ed elementi tipo), adeguamento delle espressioni regolari per i tipi predefiniti di dati, implementazione delle suddette modifiche, adeguamento alle specifiche e raggruppamento di primaryKeyType e candidateKeyType in uniqueKeyType. Aggiunta di DATALINK da SQL:2008 parte 9. Aggiunta di clobType e blobType.</p>	