

Manuale di installazione di TheMatrix

Versione 0.64 del Manuale — settembre 2014

**Consiglio Nazionale delle Ricerche
Istituto di Scienza e Tecnologie
dell'Informazione “A.Faedo”**

Autori

Massimo Coppola, Iacopo Peri, Giacomo Righetti,
Emanuele Carlini



Consiglio Nazionale delle Ricerche



Versione 0.64 del Manuale, settembre 2014

Riferita al programma **TheMatrix** versione 1.52 del settembre 2014

Note alle versione 1.52 del software, versione 0.64 del manuale — Aggiornamento dei file risultanti dall'esecuzione, aggiornamenti generali.

Note alle versione 1.23-1.25 del software, versione 0.61-0.62 del manuale — Aggiornamento di: esecuzione da riga di comando, formato dei dati SQL e CSV.

Note alle versione 1.20 - 1.22 del software, versione 0.6 del manuale — Dalla versione 1.20 è attivo il linguaggio di estrazione dati, che non è però descritto nel presente manuale. L'integrazione con i database, in corso di modifica, non include tutti i tipi di mapping definiti nel manuale.

Note alle versioni 0.5 – 1.02 del software — Per la configurazione di **TheMatrix** sono necessarie solo le funzionalità di estrazione dei dati. Le fasi di elaborazione e spedizione non sono abilitate.

Note alle versioni 0.3 – 0.4 del manuale — Il manuale contiene le informazioni necessarie alla configurazione, e non contiene il capitolo sul linguaggio di scripting.

Nota alla versione 0.41 del manuale — La versione 0.41 del manuale è una release *ad interim* legata al rilascio della prima versione di compatibilità con Microsoft SQL Server.

Contatti

Dr. Massimo Coppola massimo.coppola@isti.cnr.it

Indice

1	Introduzione a TheMatrix	1
1.1	Licenza Software e documentazione	1
1.2	Cosa è TheMatrix	2
1.2.1	Caratteristiche e Vincoli di Progettazione	4
1.2.2	Schemi di dati: IAD e LIAD	5
1.2.3	Formati dati e tecnologie software utilizzate	5
2	Manuale d'uso	7
2.1	Uso di TheMatrix	7
2.2	Opzioni di lancio	8
2.2.1	Esempio	10
2.2.2	Tool di supporto	11
3	File di configurazione	13
3.1	Accesso al DBMS	14
3.2	File Settings.xml	15
3.2.1	Sezione Version	16
3.2.2	Connessione con il DBMS – Sezione DbConnection	16
3.2.3	Configurazione dei percorsi – Sezione Path	18
3.2.4	Esempi di file settings.xml con e senza connessione DBMS	19
3.3	File mapping.xml	21
3.3.1	IadMapping	21
3.3.2	Operazioni di Join	22
3.3.3	Le diverse tipologie di mapping	22
3.3.4	VoidMapping	23
3.3.5	SimpleMapping	23
3.3.6	MultiMapping	24
3.3.7	lookupMapping	26

3.3.8	complexMapping	27
4	Risultati	29
4.1	Tipi di File di supporto	29
4.2	CSV finali	30
4.3	I file XML	30
4.4	File contenenti Query	31
A	Esempi di configurazione	33

Capitolo 1

Introduzione a TheMatrix

TheMatrix è un applicativo in grado di connettersi ad una base di dati, eseguire in maniera automatica o semiautomatica operazioni di estrazione di una parte dei dati contenuti, trasformare i dati estratti in maniera prestabilita, ed opzionalmente inviare il risultato della elaborazione come file CSV ad un sito esterno. Il presente manuale versione 0.64 si riferisce al programma TheMatrix versione 1.52.

TheMatrix è sviluppato dallo ISTI-CNR e dalla ARS Toscana nell'ambito del progetto Matrice, progetto sostenuto dal Ministero della Salute e coordinato dalla AGeNas. TheMatrix è stato progettato a partire da un insieme di tecnologie open-source (Jerboa, Neverlang) ed in base a specifici requisiti che ne rendono sicuro (dal punto di vista informatico) e lecito (dal punto di vista legale) l'utilizzo a partire dai database delle ASL campione del progetto Matrice.

1.1 Licenza Software e documentazione

Per la durata del progetto Matrice le distribuzioni prototipo del programma TheMatrix sono da considerare confidenziali e non redistribuibili dai partner del progetto.

Il programma TheMatrix è stato rilasciato con licenza GPL3 ed è disponibile per il download a questo indirizzo:

`https://github.com/hpclab/TheMatrixProject`

La documentazione, che comprende questo manuale, una guida di installazione rapida, la sintassi completa del linguaggio ed la documentazione per gli sviluppatori Java è disponibile a questo indirizzo:

`https://github.com/hpclab/TheMatrixProject-doc`

TheMatrix è sotto copyright degli autori e dello ISTI-CNR per la parte sviluppata all'interno del progetto Matrice, e dell'autore del programma Jerboa, Martijn Schuemie. Il tool Neverlang è copyright dei rispettivi autori dello ADAPT Lab dell'Università di Milano ed è stato utilizzato per la generazione dell'interprete del linguaggio. Più informazioni su Neverlang possono essere trovate su <http://neverlang.di.unimi.it/>.

Ringraziamenti Si ringraziano la D.sa Rosa Gini (Agenzia Regionale di Sanità Toscana), a cui si deve l'idea iniziale di TheMatrix, per la preziosa collaborazione nella progettazione; Martijn Schuemie (Erasmus MC, University Medical Center Rotterdam) per la collaborazione e la disponibilità a condividere il codice di Jerboa; Edoardo Vacchi e Walter Cazzola dello ADAPT Lab dell'Università di Milano, per lo sviluppo di Neverlang e la collaborazione allo sviluppo di TheMatrix; lo staff tecnico di ARS Toscana ed i colleghi Patrizio Dazzi ed Emanuele Carlini per l'aiuto e la collaborazione durante lo sviluppo di TheMatrix.

1.2 Cosa è TheMatrix

TheMatrix è un'applicazione Java che accede ad un database amministrativo *sorgente*, ne estrae un sottoinsieme dei dati, li elabora in maniera automatizzata e produce un file di risultato che viene trasferito all'esterno.

A cosa serve TheMatrix ha due funzioni fondamentali

- adattare i dati amministrativi presenti in uno specifico sito (caso tipico, il database di una ASL) ad un formato ed una rappresentazione comuni, che consentano di raccogliere e trattare tali dati in modo uniforme;
- applicare ai dati raccolti in loco algoritmi che estraggano indicatori significativi. Tali indicatori, per esemplificare, possono segnalare il riconoscimento, la cura appropriata o la probabilità di sviluppare una data patologia per gli individui di una popolazione assegnata.

Gli indicatori possono essere destinati sia all'uso locale, sia all'analisi centralizzata. Gli algoritmi che li calcolano sono il risultato del lavoro di ricercatori medici coinvolti nel progetto Matrice del M.Sal. ed in altri progetti analoghi.

Chi usa TheMatrix TheMatrix ha tre classi di utenti, con una certa intersezione di competenze e ruoli:

ricercatori ed epidemiologi i ricercatori medici preparano gli algoritmi per il calcolo degli indicatori a partire dai dati grezzi. Ogni algoritmo è codificato da uno o più programmi nel linguaggio interno di **TheMatrix**, programmi che chiameremo *script* e che possono essere distribuiti assieme al programma stesso o separatamente, in una fase successiva. Lo sviluppo degli script, sulla base dei risultati della ricerca medica e di meccanismi di validazione statistica, è normalmente uno dei passi di una lunga attività di ricerca.

analisti medici e supervisori amministrativi gli indicatori calcolati da **TheMatrix** possono essere utilizzati per la supervisione ed il monitoraggio di insiemi di pazienti e procedure di cura, con scopi di controllo amministrativo e/o medico; nell'ambito del progetto Matrice i dati generati da **TheMatrix** sono inviati ad altri tool che si occupano appunto della loro visualizzazione e della generazione di diverse tipologie di report.

operatori/responsabili dei sistemi informatici la terza categoria di utenti ha direttamente il compito di eseguire il programma **TheMatrix**, di configurarne l'accesso ai dati (tipicamente una sola volta all'installazione, a meno di release successive) ed agli script (ogni volta che viene sviluppata una nuova versione degli script), nonché di eseguire gli script stessi e fornire i risultati agli analisti e/o ai tool di generazione dei report (questo secondo passaggio può anche essere automatizzato).

Quali dati vengono usati I dati utilizzati da **TheMatrix** sono quelli corrispondenti al formato comune dei dati sanitari amministrativi definito dal Ministero della Salute contenuti nel database sorgente. **TheMatrix** è realizzato in modo da essere indipendente dalla specifica installazione locale di una ASL del database.

Come sono elaborati i dati L'estrazione e l'elaborazione sono governate da specifici script (sottoprogrammi) scritti nel linguaggio di **TheMatrix**, che definiscono nel dettaglio il tipo di analisi dei dati richiesta di volta in volta.

Che risultati sono prodotti I risultati sono tipicamente un piccolo insieme di indicatori calcolati a partire dai dati amministrativi stessi, per tutti o per un sottoinsieme dei soggetti menzionati nel database; i risultati sono in un formato (CSV) e con caratteristiche di implementazione fissate per poter essere utilizzati all'interno del progetto di ricerca Matrice.

1.2.1 Caratteristiche e Vincoli di Progettazione

Il programma TheMatrix è stato realizzato con precise funzionalità e vincoli, riassunti nel seguito.


- Capacità di interfacciamento a diversi DBMS sorgente, sia via SQL e JDBC, sia in modalità manuale, via file di testo e file CSV.
 - L’interfaccia via SQL, ove reso possibile dal DBMS di supporto, può avvenire tramite un utente dedicato che non ha accesso al database in scrittura, ed ha a disposizione una vista dedicata contenente solo le tabelle indispensabili al programma.
 - L’interfaccia per l’accesso ai dati in forma testuale può anche accedere a dati già in formato CSV, senza richiedere in questo caso il collegamento diretto ad un database.
- L’elaborazione dei dati è governata da file di script (sottoprogrammi) nel linguaggio interno di TheMatrix che descrivono i vari passi dell’elaborazione ed gli specifici dati da estrarre. Gli script sono tipicamente realizzati da un team di ricercatori nell’ambito del progetto Matrice, vengono forniti al programma TheMatrix come argomento per l’esecuzione, e producono un file di risultati che viene inviato ai ricercatori. Il contenuto dello script è ispezionabile prima dell’esecuzione e protetto da modifiche con tecniche standard di cifratura e segnature.¹
- I dati identificativi espliciti (p.es. codice fiscale) non sono mai comunicati all’esterno in forma utile: un meccanismo crittografico specifico seguito da una fase di anonimizzazione permettono di collegare i risultati ottenuti da TheMatrix nell’ambito di studi generali, ma impediscono di risalire all’identificativo originale di un qualsiasi soggetto a partire dai file risultato.
- La configurazione iniziale del programma permette di etichettare gli specifici campi che contengono dati identificativi. TheMatrix verifica che nessuno di questi campi entri a far parte del risultato senza essere anonimizzato.
- La spedizione dei file risultato, prodotto in formato CSV, può avvenire in automatico tramite canale sicuro (SSL/TLS) oppure essere effettuata manualmente dal sistemista responsabile, che ha quindi la possibilità di verificare il contenuto inviato.

¹Criptazione RSA 1024 bit. Attenzione: tale caratteristica non è attiva nella versione 1.52 di TheMatrix.

1.2.2 Schemi di dati: IAD e LIAD

Sia i dati presenti nel DBMS sorgente, sia quelli trattati da TheMatrix corrispondono ad uno schema preciso, che definisce tipo, formato e significato di ogni campo dei record del database. Tale schema comprende varie tabelle. Nel caso di TheMatrix gli schemi delle tabelle dei database localizzati presso le ASL devono poter corrispondere ad uno schema stabilito dal Ministero della Salute e dal Ministero delle Finanze, ex art. 50 DL 269/2003. Chiameremo nel seguito IAD l'implementazione di detto schema definita nel progetto Matrice.

In generale i dati contenuti nel DBMS sorgente, possono non corrispondere esattamente alla specifica IAD; il primo compito di TheMatrix è eseguire le opportune trasformazioni che riportino i dati rappresentati all'interno del database sorgente nel formato IAD. I dati contenuti nel database sorgente sono comunque un sovrainsieme dello schema IAD, eventualmente implementato in maniera tecnicamente diversa ma concettualmente equivalente. Nel seguito quindi chiameremo LIAD (Local IAD) lo schema di uno specifico database sorgente.

 Mentre lo schema IAD implementato in TheMatrix è comune, diversi database sorgente avranno in generale uno schema LIAD diverso tra loro. L'adattamento tra lo specifico LIAD e lo IAD è effettuato da TheMatrix seguendo le specifiche fornite in fase di installazione presso la ASL. A questo scopo, il responsabile dei sistemi informativi di una singola ASL ha necessità di conoscere 1) lo schema IAD e 2) il solo schema LIAD del proprio database.

La realizzazione di TheMatrix è basata sulle seguenti normative ministeriali relative al formato dei dati (vedi sezione ??).

1.2.3 Formati dati e tecnologie software utilizzate

Le tecnologie usate dall'applicativo TheMatrix sono:

- XML — eXtensible Markup Language (formato dati)
- CSV — Comma Separated Values (formato dati)
- Java (linguaggio di programmazione)
- Jerboa (Scripting language per elaborazione dati)
- Jaxb (Java Architecture for XML Binding)
- JDBC (Standard di connessione a DBMS SQL per Java)

- Neverlang (framework Java per la costruzione di Application-specific languages).

Capitolo 2

Manuale d'uso

In questo capitolo descriviamo le modalità di esecuzione, il significato e l'uso dei parametri disponibili, sia per il programma **TheMatrix** che per i tool di supporto. In generale la differenza tra l'uso su sistemi UNIX-like (inclusi LINUX, Apple OS X) e Microsoft Windows è limitata, dato che i tool JAVA sono richiamati tramite script bash (UNIX) o .bat (Windows) che si occupano dei dettagli specifici legati al sistema operativo.

Importante: in questo capitolo si parlerà di due tipi diversi di script, prestare attenzione a non confondere

- script nel linguaggio di **TheMatrix**, che specificano l'elaborazione dei dati da eseguire
- script del sistema operativo della macchina usata (script bash su sistemi Unix/Linux e derivati, script batch DOS su sistemi Microsoft Windows), che sono usati per semplificare la procedura di esecuzione di **TheMatrix**.

2.1 Uso di TheMatrix

Per eseguire l'applicativo **TheMatrix** è necessario, da shell o finestra DOS, portarsi nella directory in cui è installato e quindi digitare l'apposito script di esecuzione (corrispondente al proprio sistema operativo) e passare come parametro almeno il nome dello script di elaborazione dati da eseguire, come negli esempi seguenti.

```
cd directoryDiInstallazione
./TheMatrix nomeScript
```

UNIX

```
cd C:\TheMatrix
TheMatrixWIN nomeScript
```

Windows

In tutti i casi *nomeScript* è il nome (senza percorso, con l'estensione `.txt`) del file di script nel linguaggio **TheMatrix**. Di regola tale script deve trovarsi nella cartella *script* dell'installazione. Per convenzione, tutti gli script di **TheMatrix** hanno estensione `.txt`

È possibile passare a **TheMatrix** una serie di opzioni, oltre al nome dello script da eseguire; le opzioni modificano il normale comportamento, quelle rilevanti per l'utente sono descritte nella successiva sezione ??.

UNIX

```
./TheMatrix opzione1 opzione2 ... nomeScript
```

Windows

```
TheMatrixWIN opzione1 opzione2 ... nomeScript
```

Dopo l'avvio, al momento in cui fosse necessario creare una connessione JDBC con un database, il programma richiederà la password di accesso al DBMS. In caso di inserimento erraneo da parte dell'utente **TheMatrix** termina segnalando l'errore. Se la password è corretta, l'accesso JDBC viene effettuato per tutte le tabelle necessarie. La password **non** è registrata su disco, quindi deve essere inserita ad ogni nuova esecuzione che richieda di attivare connessioni JDBC.

L'esecuzione di **TheMatrix** produrrà alternativamente:

- uno o più file di risultati, se i dati necessari sono tutti disponibili; detti file, alla versione corrente di **TheMatrix**, si trovano nella directory *iad*
- una query da sottomettere al DBMS per ricavare (parte de) i dati mancanti.

Per maggiori informazioni al riguardo si consulti il cap. ??.

2.2 Opzioni di lancio

L'applicazione Java **TheMatrix** accetta numerosi parametri da riga di comando. Tali parametri sono anche accettati dagli script di esecuzione per Unix e Windows, vengono direttamente trasferiti all'applicazione Java e dunque si usano allo stesso modo.

La forma della riga di comando è descritta nella sezione ??, in particolare il nome dello script da eseguire, se presente, va sempre per ultimo sulla riga di comando. Alcune opzioni richiedono un parametro, che deve obbligatoriamente seguire l'opzione stessa. Se il parametro è un nome di file o di directory, seguirà la forma adottata dal sistema operativo ospite.

La tabella ?? offre una panoramica di tutti i parametri possibili a riga di comando. Una versione più dettagliata e precisa si può trovare di seguito.

comando	breve descrizione
--help	stampa un le opzioni di uso dell'applicazione
--ignoreDBconnection	disabilita l'accesso al DBMS via JDBC
--fullIADschema	seleziona una versione del modello dei dati IAD
--dry-run	esegue il controllo dello script
--scriptPath	imposta una directory differente per gli script
--iadPath	imposta una directory differente per i dati
--resultPath	imposta una directory differente per i risultati
--loglevel	imposta il livello di dettaglio dei messaggi di log
--dumpConsumerGraph	stampa il grafo dei moduli
--saveConsumerGraph	salva il grafo dei moduli su file
--dumpMappingSchema	stampa una rappresentazione degli schemi usati

Tabella 2.1: Tabella riassuntiva parametri a riga di comando di TheMatrix

--help stampa sullo schermo un breve testo che riassume le opzioni di uso dell'applicazione, quindi termina senza eseguire alcuna elaborazione.

--ignoreDBconnection disabilita l'accesso al DBMS via JDBC, anche se configurato; con questo parametro, ogni volta che un file di dati IAD non è disponibile, TheMatrix genererà la query SQL corrispondente e arresterà l'elaborazione. È utile nella fase di debug dei file di mapping per analizzare quali query vengono generate da TheMatrix prima di tentare una connessione JDBC.

--fullIADschema <integer> seleziona una delle versioni disponibili del modello dei dati IAD. Per la versione 1.52 di TheMatrix, <integer> è un intero tra 0 e 2 con significato

0 sceglie il formato IAD parziale derivato dallo schema dati ARS, e usato durante i test interni

1 sceglie il formato IAD completo definito per l'estrazione dei dati nelle ASL campione;

NOTA: TheMatrix, alla versione 1.52, usa tale opzione per default.

2 sceglie il formato IAD completo definitivo, come fissato nella fase di sperimentazione di Matrice.

--dry-run non effettuare alcuna delle elaborazioni richiesta dallo script, ma esegui il controllo sintattico e di correttezza formale dello script.

- scriptPath <directory name>** permette di impostare una directory differente da quella di default per la ricerca degli script **TheMatrix** (nota: <directory name> è un path relativo rispetto alla directory di installazione); parametro usato in generale solo per il test del programma;
- iadPath <directory name>** permette di impostare una directory differente da quella di default per la ricerca dei file di dati IAD (nota: <directory name> è un path relativo rispetto alla directory di installazione);
- resultPath <directory name>** permette di impostare una directory differente da quella di default per la memorizzazione degli output conseguente all'esecuzione di script **TheMatrix** (nota: <directory name> è un path relativo rispetto alla directory di installazione);
- loglevel <integer>** imposta il livello di dettaglio dei messaggi di log sullo schermo e nel file di log. <integer> è un valore intero compreso tra 0 (solo informazioni essenziali) e 3 (massimo livello di dettaglio);
- dumpConsumerGraph** scrive il grafo dei moduli sulla console e su log, usato principalmente dagli sviluppatori;
- saveConsumerGraph <filename>** salva il grafo dei moduli sul file passato per argomento, usato principalmente dagli sviluppatori;
- dumpMappingSchema** scrive sul terminale una rappresentazione degli schemi usati dal programma (coerentemente con il valore dell'opzione **--fullIADschema**, se presente); tale rappresentazione è alla versione 1.52 di **TheMatrix** un file XML che può essere usato come template per il file di mapping dell'applicazione; ciò può essere utile a verificare che il file di mapping contenga tutte le definizioni necessarie al programma. Note:
1. con questa opzione il programma termina immediatamente dopo avere stampato gli schemi, senza eseguire alcuno script;
 2. per usare il template prodotto al momento è necessario fare copia e incolla dal terminale.

2.2.1 Esempio

Nel seguito un esempio di lancio di **TheMatrix** da riga di comando. Stiamo avviando l'esecuzione di **TheMatrix** con lo script "script_diabete_130419.txt", richiedendo di effettuare

la sola verifica di correttezza dello script (e non l'elaborazione), utilizzando la definizione 1 dello schema IAD (versione completa ad Aprile 2013).

```
./TheMatrix --dry-run --fullIADschema 1 script_diabete_130419.txt
```

UNIX

```
TheMatrixWIN --dry-run --fullIADschema 1 script_diabete_130419.txt
```

Windows

2.2.2 Tool di supporto

Per eseguire il tool di supporto a **TheMatrix** MetaXMLCreator, che abilita come fonte di dati un csv generato manualmente, è necessario digitare da shell:

```
./MetaXMLCreator nomefile.csv
```

UNIX

```
MetaXMLCreatorWIN nomefile.csv
```

Windows

Il comando va eseguito dalla directory di installazione (normalmente **TheMatrix** o **C:\TheMatrix**). Qui *nomeFile.csv* è il nome del file CSV importato, di cui si vuole creare il meta descrittore.

Il file è indicato senza percorso, poiché deve trovarsi nella directory **iad** ; il programma assume che abbia estensione **.csv** anche se questa non fosse indicata.

Il tool analizza il file CSV indicato, creando un file XML che **TheMatrix** utilizza per effettuare i controlli di validità. Senza il meta descrittore relativo al file CSV **TheMatrix** non considera valido il file CSV ricavato, quindi ogni volta che viene aggiunto manualmente un file CSV è indispensabile eseguire MetaXMLCreator.

Una descrizione più dettagliata del procedimento è data alla sezione ??.

Capitolo 3


File di configurazione


L'applicazione **TheMatrix** usa due file di configurazione in formato XML:

settings.xml è un file di configurazione globale per impostare i percorsi necessari al funzionamento di **TheMatrix**, nonché tutte le informazioni generali necessarie all'esecuzione per interagire con la macchina ospite.

mapping.xml è un file di configurazione specifico per il mapping, ovvero indica la corrispondenza tra i campi del database sorgente ed i campi dello schema nazionale IAD. Per semplicità chiameremo questo file sempre nello stesso modo, anche se il nome usato può essere modificato dall'utente agendo sul file **settings.xml**,

Si assume che il sistemista incaricato della configurazione iniziale di **TheMatrix** abbia sufficiente familiarità con il formato XML e sappia utilizzare un editor di testo per modificare i file menzionati. Nel seguito del capitolo descriveremo il significato dei vari tag presenti nei due file e come utilizzarli per adattare **TheMatrix** allo specifico sistema su cui è installato.

 I file di configurazione possono contenere informazioni critiche per la sicurezza (p.es. chiavi crittografiche o password) o per le funzionalità del programma (p.es. informazioni essenziali sulla struttura del database sorgente). Perciò si consiglia che i pieni diritti di accesso ai file di configurazione siano strettamente limitati all'amministratore di sistema. L'implementazione di **TheMatrix** assume solo che l'utente che esegue **TheMatrix** abbia i necessari diritti di lettura su questi file.

 Nel caso specifico che si inseriscano informazioni critiche per la sicurezza (e.g. chiavi crittografiche o password nei file di configurazione), è cura del sistemista far sì che tali file siano leggibili solo durante l'esecuzione di **TheMatrix**, e non prima o dopo (p.es.

tramite uso dei bit di protezione `s` su sistemi Unix).

3.1 Accesso al DBMS

L'accesso ai dati presenti nel DBMS da parte di **TheMatrix** avviene secondo alcune assunzioni fondamentali che riportiamo nella presente sezione.

1. **TheMatrix** apre una connessione con il DBMS tramite un driver JDBC. Il database di supporto deve essere compatibile con lo standard JDBC. Per tale connessione deve essere in generale creato un utente apposito nel database, utente che verrà poi comunicato a **TheMatrix** per mezzo dei file di configurazione.
2. Il database contenente i dati necessari a ricostruire lo schema di dati IAD, in generale, sarà organizzato in diverse tabelle dati. Per la versione 1.52 di **TheMatrix** è possibile specificare un solo database a cui accedere per caricare le diverse tabelle. In seguito sarà reso possibile specificare differenti connessioni e server per le sorgenti necessarie ad ogni tabella IAD. Non sarà presumibilmente supportata la ricostruzione di una singola tabella IAD a partire da due database distinti.
3. L'accesso ai dati da parte di **TheMatrix** è necessario in sola lettura. È anzi fortemente consigliato, in base al principio di minimo privilegio necessario, che nell'installazione di **TheMatrix** alla creazione dell'utente del database dedicato a **TheMatrix**, a tale utente **non siano attribuiti** diritti in scrittura sulle tabelle dati.
4. **TheMatrix** può accedere indifferentemente sia ad tabelle materializzate del DBMS, sia ad una vista distinta dal vero schema del DBMS. L'amministratore del DBMS può scegliere la soluzione operativamente più semplice nel caso specifico.
5. Le impostazioni stabilite dall'amministratore del DBMS devono essere poi riportate nel file settings, di cui alla successiva sezione ??.
6. Nel caso **TheMatrix** non abbia a disposizione una connessione JDBC con il DB, **TheMatrix** genererà il testo della query senza eseguirla, salvandola in un file di testo nella directory queries (vedi tag **queries**). L'amministratore della base dati deve provvedere manualmente a
 - (a) eseguire sul database la query SQL generata da **TheMatrix**
 - (b) salvare i dati ottenuti dalla query in un file separato e convertito in formato CSV

- (c) copiare il file CSV nella directory dei file *IAD*, cioè quella specificata nel tag di configurazione `iad` descritto alla sezione ??
- (d) eseguire il tool di supporto descritto alla sezione ??
- (e) solo a questo punto avviare nuovamente **TheMatrix**, che preleverà i dati necessari dal file appena creato senza tentare di aprire una connessione con il DB.

Va notato che a seconda della configurazione di **TheMatrix** e dello script usato, le query necessarie all'esecuzione di uno script per un particolare dataset possono essere più di una; esse saranno tutte contenute nello stesso file di testo, e per ognuna l'amministratore dovrà eseguire tutti i passi da a) a d) prima di passare all'esecuzione di **TheMatrix**.

***Nota bene:** poiché la versione 1.52 di **TheMatrix** è destinata al test di compatibilità con i server delle ASL campione, questo specifico procedimento non è operativo.*


7. Il procedimento di cui al punto precedente opera fornendo in modalità manuale a **TheMatrix** una serie di file di supporto in formato CSV (file che **TheMatrix** è in grado di generare da solo avendo a disposizione una connessione JDBC correttamente configurata). Per il contenuto e le convenzioni relative a detti file si faccia riferimento al capitolo ?? ed alla sezione ??.

3.2 File Settings.xml

Il file di configurazione principale è un file XML all'interno della directory radice dell'applicazione.

```
TheMatrix/  
├── settings.xml  
└── ...
```

All'interno di questo file sono espresse, attraverso l'uso di tag XML, tutte le informazioni della configurazione di **TheMatrix**, necessarie per l'accesso ai vari tipi di risorse usate (informazioni per la connessione al database, directory, valori di parametri dell'esecuzione etc.).

 I file di configurazione contengono informazioni fondamentali per la corretta esecuzione del programma; tali informazioni devono essere inserite all'atto della prima installazione di **TheMatrix** con l'aiuto di un utente esperto o di un sistemista. È fortemente sconsigliato modificare manualmente il contenuto dei singoli tag o del loro contenuto prima

dell'esecuzione di script. Si assume che l'utente che esegue **TheMatrix** abbia i necessari diritti di lettura sul file nel momento in cui **TheMatrix** è in esecuzione.

Il tag principale che racchiude tutta la struttura XML è:

```
<theMatrix>
...
</theMatrix>
```

Al suo interno sono presenti 3 sezioni principali che servono a definire proprietà diverse.

3.2.1 Sezione Version

Questo tag serve a definire la versione dell'applicazione **TheMatrix** che si sta installando.

```
<version> ... </version>
```

Il valore di questo tag viene usato per il controllo del versionamento dei *dataset* in ingresso e per la generazione dei risultati di una esecuzione. **TheMatrix** decide la compatibilità degli script da eseguire con la versione del programma installata e con i dataset precedentemente estratti anche tenendo conto di questo tag.


3.2.2 Connessione con il DBMS – Sezione DbConnection

Questa sezione del file di configurazione è relativa alla gestione delle informazioni necessarie per la connessione con il database sorgente, attraverso cui richiedere i dati da elaborare. È definita dal tag seguente:

```
<dbConnection>
...
</dbConnection>
```

Il tag è opzionale, perché la connessione al DBMS potrebbe non essere presente.

All'interno di questo tag sono definite tutte le proprietà che formano la stringa di connessione necessaria al driver JDBC per interfacciarsi con il DBMS. Le informazioni necessarie a JDBC per identificare questa connessione sono descritte nel file *tnsnames.ora* all'interno la configurazione del DBMS stesso.

 L'istanza del database e l'utente con il quale accedere al DBMS devono essere già preesistenti e configurate, vedi ???. Le informazioni espresse dentro questo tag XML servono solo a utilizzare connessioni già impostate precedentemente. Eventuali cambiamenti della configurazione del DBMS (p.es. creazione di un utente o di una vista) sono compiti del

sistemista.

I tag descritti in questa sezione sono tutti obbligatori (senza le informazioni in essi contenute non è possibile eseguire correttamente TheMatrix). Al suo interno troviamo:

```
<dbConnection>
  <serverName> ... </serverName>
  <portNumber> ... </portNumber>
  <sid> ... </sid>
  <user> ... </user>
  <type> ... </type>
</dbConnection>
```

dove:

serverName specifica l'indirizzo di connessione al DBMS. As esempio:

`oracleDbms.ars.sanita.it`

portNumber specifica la porta sul quale stabilire la connessione con il DBMS. Per esempio, nel caso di un DBMS Oracle la porta di default è la 1521.

sid specifica l'istanza del database con il quale si vuole instaurare la connessione. È una stringa corrispondente al nome dell'istanza.


user definisce l'username dell'utente con il quale accedere al DBMS.

type definisce il tipo di DBMS con il quale collegarsi. Serve a utilizzare il corretto tipo di connessione in fase di collegamento. I possibili valori di questo tag al momento sono

oracle Oracle Database versione 10g e successive¹;

mysql MySQL;

sqlserver Microsoft SQL Server versioni² 2000, 2005 e 2008.

 Il tag `dbConnection` è definito opzionale poichè potrebbe non essere disponibile una connessione con un DBMS. Per dichiarare la mancanza di disponibilità verso un database è necessario escludere questo tag e tutti i tag figli dal file XML di configurazione. Se il tag `dbConnection` è presente, i tag figli sono obbligatori.

¹versioni successive non testate

²compatibilità non testata

3.2.3 Configurazione dei percorsi – Sezione Path

La sezione relativa alla definizione dei percorsi di sistema su cui mappare i dati e i file necessari viene definita da questa sezione all'interno del file di configurazione. Il tag principale che racchiude tutte le singole proprietà è:

```
<path>
...
</path>
```

Durante l'esecuzione di **TheMatrix** vengono letti e generati numerosi file di natura differente; per ogni tipo di file, deve essere definito un path di ricerca nel filesystem³. I path da specificare in questa sezione sono i seguenti:

```
<path>
  <results> ... </results>
  <iad> ... </iad>
  <mapping> ... </mapping>
  <scripts> ... </scripts>
  <queries> ... </queries>
  <lookuptables> ... </lookuptables>
</path>
```

dove:

results definisce il percorso relativo ai file generati da **TheMatrix** come output;

iad definisce il percorso relativo ai file ottenuti dalla lettura del DBMS. Viene eseguito un controllo di versione per evitare l'uso di file modificati o obsoleti; i file che non superano il controllo sono ignorati e, se necessari all'elaborazione, vengono rigenerati. Nella directory specificata vengono generate coppie di file, una per ogni file ottenuto del DBMS; ogni coppia è costituita da un file dati in formato CSV e da un file XML contenente i metadati relativi al primo file. Il formato del file di metadati è descritto più in dettaglio nel cap. ??

mapping definisce il percorso in cui trovare il file di configurazione mappings.xml, dove è definito il mapping IAD/LIAD (vedi ??)

scripts definisce il percorso in cui trovare i file di script

³Nella versione installata presso le ASL, all'avvio dell'applicazione viene eseguito un controllo di validità dei path specificati nel file di configurazione.

queries definisce il percorso relativo ai file generati contenenti le query SQL da eseguire via JDBC o manualmente (vedi capitolo ??)

lookuptables definisce il percorso relativo ai file contenenti le tabelle di lookup.

3.2.4 Esempi di file settings.xml con e senza connessione DBMS

Esempio di settings.xml con connessione DBMS

```
<theMatrix>
  <version>0.1</version>
  <dbConnection>
    <serverName>oracleDBMS.arst.sanita.it</serverName>
    <portNumber>1521</portNumber>
    <sid>arst</sid>
    <user>dbmsUser</user>
    <type>oracle</type>
  </dbConnection>
  <path>
    <results>result_folder</results>
    <iad>iad_folder</iad>
    <mapping>mapping_folder/mapping.xml</mapping>
    <scripts>script_folder</scripts>
    <queries>queries_folder</queries>
    <lookuptables>lookuptable_folder</lookuptables>
  </path>
</theMatrix>
```

Esempio di settings.xml senza connessione DBMS

```
<theMatrix>
  <version>0.1</version>
  <path>
    <results>result_folder</results>
    <iads>iad_folder</iads>
    <mappings>mapping_folder/mapping.xml</mappings>
    <scripts>script_folder</scripts>
    <queries>queries_folder</queries>
```

```
</path>  
</theMatrix>
```


3.3 File mapping.xml

Come spiegato nella sezione ??, LIAD identifica il formato dati di un database locale ad un'azienda sanitaria locale; IAD invece identifica il formato dati a livello nazionale.

Il processo di mapping in **TheMatrix** rappresenta la traduzione a livello di struttura e, almeno potenzialmente, a livello di contenuto di un insieme di dati, originariamente espressi in un formato LIAD. L'obiettivo della trasformazione è ricondurre i dati alla rappresentazione secondo lo schema IAD per le successive elaborazioni.

Il file *mapping.xml*, che viene scritto dall'amministratore di sistema nella fase di installazione di **TheMatrix**, descrive come mappare uno specifico dataset LIAD nello schema IAD.

La descrizione si divide in una parte globale, che indica in che modo i dati sono estratti dal DBMS sorgente, ed una parte analitica, in cui per ogni singolo campo IAD viene definito come esso viene ricavato a partire da uno o più campi dello schema LIAD.

Il file *mapping.xml* utilizza un formalismo derivato dall'XML. Nel seguito sono presentati gli elementi XML che compongono il file stesso, assieme a semplici esempi.

3.3.1 IadMapping

E' il root element del documento XML di mapping.

```
<iadMapping>
  ...
</iadMapping>
```

Tutto il contenuto deve essere racchiuso tra questi due elementi.

Dataset

Contiene le informazioni necessarie a mappare uno specifico dataset IAD. Il nome del dataset è esprimibile tramite l'attributo *name*. In uno stesso IadMapping in generale sono contenuti vari Dataset consecutivi.

```
<dataset name="nomeDatasetDaMappare">
  ...
</dataset>
```

3.3.2 Operazioni di Join

All'interno dell'elemento *dataset* deve essere specificata la clausola di join necessaria per riunire informazioni appartenenti a tabelle distinte. Questo avviene mediante gli elementi *joinClause* e *joinName*.

```
<dataset name="nomeDatasetDaMappare">
  <joinClause>SPF1 JOIN SPF2 ON SPF1.ID_SPF1=SPF2.ID_SPF1</joinClause>
  <joinName>mionomefile</joinName>
</dataset>
```

Il tag *joinClause* è utilizzato per costruire una query SQL, quindi è necessario che sia corretto e privo di caratteri estranei; un errore comune è quello di chiudere il tag alla riga successiva, in questo modo se inserisce un fine linea spurio all'interno della query SQL.

Importante: se una operazione di Join non è necessaria, perché i dati sono contenuti in un'unica tabella nel database locale, allora il tag dovrà semplicemente contenere il nome della tabella.

Il contenuto del tag *joinName* è il nome del file CSV ausiliario in cui, se necessario eseguire la query, viene scaricato il risultato della query. Notare che, nel caso l'integrazione al DBMS non sia usata, tale file deve essere preparato dal sistemista eseguendo manualmente la query e convertendo il risultato in CSV (si vedano la sezione ?? ed il capitolo ?? per maggiori dettagli al riguardo). Il tag *joinName* deve essere presente e contenere un nome di file valido per la vostra macchina e sistema operativo; in generale è sufficiente scegliere un nome derivato da, o uguale al nome del dataset IAD di cui si definisce il mapping.

Obsoleto il tag *connectionString* usato in versioni preliminari di *TheMatrix* è da considerarsi obsoleto.

🔗 Le join si esprimono con l'operatore standard join piuttosto che con l'utilizzo di where; si delega all'ottimizzatore DBMS il compito di rimuovere dalla condizione di join eventuali tabelle che non fossero necessarie per la query richiesta.

3.3.3 Le diverse tipologie di mapping

Successivamente compaiono i descrittori di mapping dei singoli attributi del dataset. Questi possono essere di quattro tipi:

- voidMapping
- simpleMapping

- multiMapping
- lookupMapping
- complexMapping

✎ Questi elementi non sono mutuamente esclusivi, possono essere usati liberamente per definire il mapping di differenti campi IAD. È però un errore definire più tag mapping (anche uguali tra loro) che abbiamo come risultato uno stesso campo IAD.

Indipendentemente dal tipo di mapping, è possibile marcare un attributo come dato sensibile impostando a *true* l'attributo XML *isSensible*.

Ad esempio nel caso del *simpleMapping*:

```
<simpleMapping name="nomeAttributo" isSensible="true">
    ...
</simpleMapping>
```

3.3.4 VoidMapping

Questo mapping in realtà indica che uno specifico campo del modello IAD non può essere ricavato dal modello LIAD, dunque verrà lasciato vuoto o riempito con un valore di default. In generale ciò non dovrebbe avvenire, questo tag è stato definito per permettere di costruire per gradi un mapping funzionante, ignorando i campi il cui mapping non è ancora stato definito, ma generando comunque il modello dati IAD corretto.

È possibile indicare il valore di default da assegnare a questo campo con il tag opzionale *defaultValue*; se questo non è indicato il campo sarà impostato al valore speciale *missing*. Qui sotto, due esempi rispettivamente senza valore di default, e con valore di default la stringa “Non Eseguito”.

```
<voidMapping name="esempioNomeAttributo"> </voidMapping>

<voidMapping name="esempioNomeAttributo2">
<defaultValue>Non Eseguito</defaultValue>
</voidMapping>
```

3.3.5 SimpleMapping

Questo particolare mapping rappresenta la tipologia più semplice in cui la corrispondenza tra elemento da mappare ed elemento mappato è 1 a 1. Concretamente, questo

significa che un attributo IAD, specificato tramite l'attributo XML *name*, è mappato su un singolo attributo LIAD.

Gli elementi che identificano l'attributo e la tabella sorgente sono *sourceTable* e *sourceAttribute* :

```
<simpleMapping ... >
  <sourceTable >...</sourceTable>
  <sourceAttribute >...</sourceAttribute>
</simpleMapping>
```

RecodingTable

All'interno del tag *simpleMapping* è presente anche un tag utile a definire una tabella di ricodifica valori associata all'attributo IAD da mappare.

```
<simpleMapping ... >
  ...
  <recodingTable>
    ...
  </recodingTable>
</simpleMapping>
```

Questa tabella è utilizzata per specificare regole di trasformazione per il contenuto dell'attributo IAD di cui si sta definendo il mapping. A fini esemplificativi è mostrato un esempio di un possibile mapping dei valori dell'attributo sesso.

```
<recodingTable>
  <value src="M" dest="M"/>
  <value src="m" dest="M"/>
  <value src="F" dest="F"/>
  <value src="f" dest="F"/>
</recodingTable>
```

Laddove sia necessario esprimere ricodifiche non banali o costituite da un significativo quantitativo di righe si suggerisce l'utilizzo di un *LookupMapping* (vedi sezione ??).

3.3.6 MultiMapping

Rappresenta un mapping 1-n. Questo significa che un attributo IAD si mappa su molteplici attributi LIAD appartenenti potenzialmente a più di una tabella su DBMS. L'attributo IAD da mapparsi è specificato tramite l'attributo XML *name*.

Gli attributi *sourceTable* e *sourceAttribute* indicano rispettivamente la tabella e l'attributo LIAD all'interno di questa sul quale l'attributo IAD è mappato.

A differenza di quanto avviene nel caso di mapping semplice, per il multi mapping è necessario indicare quale funzione di aggregazione debba utilizzare **TheMatrix** per unire i valori degli attributi LIAD. Le funzioni di aggregazione accettabili sono elementari e pre-stabilite dal prototipo **TheMatrix**; al momento (versione 1.52) sono supportate le seguenti funzioni:

- concat
- concatUppercase
- sum
- multiply

```
<function>nomeFunzioneAggregazione</function>
```

Successivamente sono riportati almeno due mapping. La presenza di un solo mapping è considerarsi errorea (da un punto di vista pratico, indica semmai che un **SimpleMapping** debba essere utilizzato invece del **MultiMapping**). Si faccia particolare attenzione al fatto che nel **MultiMapping** non è permesso esprimere tabelle di ricodifica. Laddove queste siano necessarie, si faccia uso del *ComplexMapping* (si veda a tal proposito la sezione ??).

```
<multiMapping>
...
<mapping>
  <sourceTable>table1</sourceTable>
  <sourceAttribute>attribute1</sourceAttribute>
</mapping>
<mapping>
  <sourceTable>table2</sourceTable>
  <sourceAttribute>attribute2</sourceAttribute>
</mapping>
...
</multiMapping>
```

3.3.7 lookupMapping

Rappresenta un mapping che utilizza tabelle di lookup. Una tabella di lookup non è altro che una tabella di database in cui una chiave primaria può essere utilizzata come meccanismo per ricavare una serie di dati accessori. Ad esempio, una tabella di lookup per medicinali potrà avere come chiave primaria un codice di omologazione, e come campi accessori le caratteristiche rilevanti del farmaco (principio attivo, eccipienti etc.).

Le tabelle di lookup disponibili nella versione 1.52 di **TheMatrix** sono rappresentate come file e memorizzate in una directory apposita.

Il `lookupMapping` permette di utilizzare un campo LIAD come chiave in una tabella di lookup già esistente, per ottenere il valore degli attributi accessori legati al particolare valore che il campo chiave assume via via. In questo modo si ottengono un certo numero di campi necessari per la creazione dello schema IAD.

Il `lookupMapping` si limita ad applicare la tabella di lookup, e non prevede alcuna altra trasformazione dei dati.

```
<lookupMapping name="attributoRisultatoDiLookup"
               inverseLookup="true">
```

Mediante l'attributo *inverseLookup* si può disabilitare la traduzione inversa in SQL. Lasciare a *true* se non si sa il significato preciso.

```
<lookupMapping name="...">
  <lookupTable>tableName</lookupTable>
  <lookupAttribute>attributeName</lookupAttribute>
  <mapping>
    <sourceTable>tableName</sourceTable>
    <sourceAttribute>attributeName</sourceAttribute>
  </mapping>
</lookupMapping>
```

Con l'elemento *lookupTable* si specifica a quale tabella di lookup fare riferimento. Con l'elemento *lookupAttribute* invece si riferisce il particolare attributo all'interno della tabella.

Con l'elemento *mapping* si indica la sorgente SQL della chiave di cui va fatto il lookup. Per ora assumiamo che la chiave non sia soggetta ad un'ulteriore decomposizione, ovvero il campo SQL sorgente della chiave è unico.

3.3.8 complexMapping

Rappresenta un mapping complesso, che prevede un rapporto 1-n e fa uso di tabelle di lookup.

```
<complexMapping name="attributoMultiplo" isSensible="true"
                inverseLookup="false">
    ...
</complexMapping>
```

Il mapping complesso prevede l'utilizzo di una funzione di mapping come nel caso del multi mapping. I parametri sono passati alla funzione nell'ordine in cui sono nel file XML, e devono essere nel numero corretto; in tutto almeno 1.

```
<complexMapping name="attributoMultiplo" isSensible="true"
                inverseLookup="false">
    <function>nomeFunzioneAggregazione</function>
    ...
</complexMapping>
```

Il mapping di tipo complesso prevede anche la possibilità di utilizzare tabelle di lookup equivalenti a quelle del lookup mapping. In questo caso le tabelle sono contenute nel tag *lookupOperation*.

Al suo interno, con *lookupTable* e *lookupAttribute* sono rispettivamente indicate la tabella di lookup usata e la colonna all'interno di detta tabella.

```
<lookupOperation>
    <lookupTable>...</lookupTable>
    <lookupAttribute>...</lookupAttribute>
    ...
</lookupOperation>
```

Con l'elemento *mapping* si indica la sorgente SQL della chiave di cui va fatto il lookup. Per ora assumiamo che la chiave non sia soggetta ad un ulteriore decomposizione, ovvero che il campo SQL sorgente della chiave sia unico.

```
<lookupOperation>
    ...
    <mapping>
        <sourceTable>tableName</sourceTable>
        <sourceAttribute>attributeName</sourceAttribute>
    </mapping>
```

```
</lookupOperation>
```

Infine, un `complexMapping` può specificare ulteriori operazioni di mapping semplici, analoghe al caso del `simpleMapping`, che si applicano tutte contemporaneamente ai diversi attributi della sorgente dati.

```
<complexMapping name="attributoMoltiplo" isSensible="true"
                inverseLookup="false">
    ...
    <mapping>
        <sourceTable>table1 </sourceTable>
        <sourceAttribute>attribute1 </sourceAttribute>
    </mapping>
    <mapping>
        <sourceTable>table2 </sourceTable>
        <sourceAttribute>attribute2 </sourceAttribute>
    </mapping>
</complexMapping>
```


Capitolo 4


Risultati

L'applicativo TheMatrix produce nell'ambito dei diversi scenari d'utilizzo due categorie di risultati: file di supporto e file relativi all'esecuzione di uno script.

4.1 Tipi di File di supporto

All'atto dell'esecuzione di uno script TheMatrix, il programma individua l'insieme di informazioni necessarie all'esecuzione. Queste sono file CSV in formato IAD che devono essere presenti all'interno della cartella specificata nel path inserito nel file di configurazione (file `settings.xml` tag `iad`). Laddove uno o più file fossero non disponibili o corrotti, l'applicativo gestisce questo evento in due diversi modi.

- La prima soluzione è stabilire una connessione alla sorgente dati DBMS ottenendo i dati mancanti in formato CSV (vedi sezione ??). Questa procedura crea il primo tipo di file di supporto che sono automaticamente salvati all'interno della cartella *iad*. Nella stessa, per ogni file CSV è definito un descrittore per i metadati del file CSV relativo (vedi sezione ??).
- Invece, nel caso in cui non fosse possibile stabilire una comunicazione col DBMS, TheMatrix esamina lo script sorgente generando l'insieme di query SQL la cui esecuzione è necessaria per ottenere i CSV mancanti. Questa tipologia di file è salvata all'interno della cartella *queries*, come descritto nella sezione ??

 Le query SQL devono essere eseguite manualmente dall'amministratore del DBMS. I risultati ottenuti devono essere convertiti in formato CSV e salvati nella cartella *iad*. Per eseguire correttamente l'applicativo è necessario utilizzare il tool di supporto *MetaXml-Creator* (vedi capitolo ??) al fine di creare un meta descrittore del file CSV importato

che TheMatrix utilizzerà durante l'esecuzione per effettuare i controlli di validità.

4.2 CSV finali

Il risultato conclusivo dell'esecuzione di TheMatrix comporta la produzione di file nella cartella *results*. Il numero di file dipende dal numero di moduli di output definiti nello script eseguito. In dettaglio per ogni modulo di output (che supponiamo abbia nome *modulo_output*) vengono generati i seguenti file:

- *modulo_output.csv* che contiene i risultati dell'elaborazione ed il cui schema dipende dallo schema del modulo di output.
- *modulo_output.xml* contiene i metadati relativi al file CSV generato.

E' importante sottolineare che TheMatrix aggiunge automaticamente un modulo di output allo script se non ne è stato specificato alcuno. In tal caso i file generati avranno prefisso *Script_<nome-file-script>_test*.

4.3 I file XML

I file XML generati da TheMatrix contengono i metadati relativi ad un generico file CSV. Le informazioni contenute sono:

- *theMatrixVersion*: La versione di TheMatrix.
- *checksumLight*: Un valore numerico utilizzato per il controllo d'integrità leggero dei dati.
- *checksumHard*: Un valore numerico utilizzato per il controllo d'integrità completo dei dati.
- *timestamp*: Marca temporale associata al file dei dati.
- *json*: Serializzazione in formato JSON¹ dello schema dei dati contenuti nel file CSV.

Le precedenti informazioni si mappano nei corrispondenti tag della seguente struttura XML:

¹<http://json.org/>

```

<csvDescriptor>
  <theMatrixVersion>...</theMatrixVersion>
  <checksumLight>...</checksumLight>
  <checksumHard>...</checksumHard>
  <timestamp>...</timestamp>
</csvDescriptor>

```

4.4 File contenenti Query

Supponendo che lo script utilizzato abbia bisogno della tabella OUTPAT, nella directory *queries* verrà generato un file con un nome analogo a

```
queryset-OUTPAT-0.1-0.5-Mon Mar 05 20:11:06 CET 2012.txt
```

formato a partire dal nome del dataset richiesto (OUTPAT), il nome dello script che richiede il dataset (in questo caso “0.1”, uno script di test interno), la versione di TheMatrix (“0.5”) e la data corrente.

Il contenuto del file sarà una query SQL analoga al seguente esempio. All’interno del file, al posto di tablename dell’esempio si troverà il nome del file CSV in cui TheMatrix andrà a cercare il risultato della query eseguita manualmente.

Esempio di Query generata da TheMatrix

```

tableName = OUTPAT_JOIN,
query = select SPECIALI, CODPRES, DATAFINE, DATAINI, QUANTUNI,
REPARTO, AZIENDA, NUMPRESTEFF, IDARS, REGAZIE
from FLUSSI.U_SPA1_EXT JOIN FLUSSI.U_SPA2_EXT
ON FLUSSI.U_SPA1_EXT.ID_SPA2_ARSNEW=FLUSSI.U_SPA2_EXT.ID_SPA2_ARSNEW

```


Appendice A

Esempi di configurazione

Per alcuni esempi del file `settings.xml` si veda la sezione ???. Nel seguito riportiamo un esempio di mapping contenente due dataset, rispettivamente realizzati senza e con una sola operazione di JOIN.

✎ Per motivi di leggibilità alcune righe del file di esempio sono spezzate in più righe del testo. Si ricorda però che nel file di mapping il testo contenuto tra due tag aperto/chiuso **non** deve contenere ritorni a capo (p.es. il testo della JOIN della tabella DRUG deve essere su un'unica riga).

```
<iadMapping>
  <!-- mapping of PERSON -->
  <dataset name="PERSON">
    <joinName></joinName>
    <joinClause></joinClause>
    <simpleMapping name="BIRTH_LOCATION_CONCEPT_ID">
      <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
      <sourceAttribute>COMNASC</sourceAttribute>
    </simpleMapping>
    <simpleMapping name="DATE_OF_BIRTH">
      <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
      <sourceAttribute>DATANASC</sourceAttribute>
    </simpleMapping>
    <simpleMapping name="ENDDATE">
      <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
      <sourceAttribute>ASSDACES</sourceAttribute>
    </simpleMapping>
    <simpleMapping name="ENDDATE_GP">
```

```

    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>DATREVOC</sourceAttribute>
</simpleMapping>
<simpleMapping name="GENDER_CONCEPT_ID">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>SEXU</sourceAttribute>
</simpleMapping>
<simpleMapping name="LHU_LOCATION_CONCEPT_ID">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>USLRESU</sourceAttribute>
</simpleMapping>
<simpleMapping name="LHU_PROVIDER_CONCEPT_ID">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>USLINVIA</sourceAttribute>
</simpleMapping>
<simpleMapping name="LOCATION_CONCEPT_ID">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>COMRESU</sourceAttribute>
</simpleMapping>
<simpleMapping name="PERSON_ID" isSensible="true">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>IDARS</sourceAttribute>
</simpleMapping>
<simpleMapping name="REG_CONCEPT_ID">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>REGRESU</sourceAttribute>
</simpleMapping>
<simpleMapping name="STARTDATE">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>ASSDECOR</sourceAttribute>
</simpleMapping>
<simpleMapping name="STARTDATE_GP">
    <sourceTable>FLUSSI.U_ATA_EXT</sourceTable>
    <sourceAttribute>DATSCELT</sourceAttribute>
</simpleMapping>
</dataset>

<!-- mapping of DRUG -->
<dataset name="DRUG">

```

```
<joinName>DRUG_JOIN</joinName>
<joinClause>FLUSSI.U_SPF1_EXT JOIN FLUSSI.U_SPF2_EXT
ON FLUSSI.U_SPF1_EXT.ID_SPF1_ARSNEW=FLUSSI.U_SPF2_EXT.ID_SPF1_ARSNEW
</joinClause>
  <simpleMapping name="DRUG_EXPOSURE_START_DATE">
    <sourceTable>FLUSSI.U_SPF2_EXT</sourceTable>
    <sourceAttribute>DATAERO</sourceAttribute>
  </simpleMapping>
  <simpleMapping name="LHU_PROVIDER_CONCEPT_ID">
    <sourceTable>FLUSSI.U_SPF1_EXT</sourceTable>
    <sourceAttribute>USLFARMA</sourceAttribute>
  </simpleMapping>
  <simpleMapping name="NUMBER_OF_BOXES">
    <sourceTable>FLUSSI.U_SPF2_EXT</sourceTable>
    <sourceAttribute>NUMFARM</sourceAttribute>
  </simpleMapping>
  <simpleMapping name="PERSON_ID" isSensible="true">
    <sourceTable>FLUSSI.U_SPF1_EXT</sourceTable>
    <sourceAttribute>IDARS</sourceAttribute>
  </simpleMapping>
  <simpleMapping name="PRODUCT_CODE">
    <sourceTable>FLUSSI.U_SPF2_EXT</sourceTable>
    <sourceAttribute>CODFARM</sourceAttribute>
  </simpleMapping>
  <simpleMapping name="REG_PROVIDER_ID">
    <sourceTable>FLUSSI.U_SPF1_EXT</sourceTable>
    <sourceAttribute>REGFARMA</sourceAttribute>
  </simpleMapping>
</dataset>
</iadMapping>
```