Università degli Studi di Modena e Reggio Emilia
Facoltà di Ingegneria “Enzo Ferrari” di Modena

Corso di Laurea in Ingegneria Informatica (270/04)

Sistemi open source di integrazione di dati: MOMIS e Pentaho Data Integration a confronto

Relatore:
Dott. Ing. Laura Po

Candidato:
Marco Maria Santese

Anno Accademico 2011/2012

Indice

INTRODUZIONE

Molte aziende hanno necessità di lavorare con sorgenti dati spesso non omogenee , basti pensare a due aziende che devono unificare i loro Database, oppure all'ambito della BioInformatica (Analisi Dna-Rna, dati molecolari, ecc); per questo motivo sono nati vari strumenti per l'integrazione. In questa relazione si analizzeranno gli aspetti più concreti e le differenze d'uso di due sistemi di integrazione dati open source : **MOMIS** (**M**ediator envir**O**nment for **M**ultiple **I**nformation **S**ources) , software sviluppato dalla collaborazione del DBGroup e DataRiver, rispettivamente gruppo di ricerca nelle Basi di Dati e una spin-off dell'Università di Modena e Reggio Emilia e **PENTAHO Data Integration (PDI)** , strumento di Integrazione di Dati basato sul progetto "Kettle" e legato al mondo della Business Analytics.

I due Tools in questione si basano su "filosofie" di integrazione di dati diverse: se MOMIS ha come punto di forza il suo funzionamento semi-automatico, basato sull'integrazione semantica delle informazioni, Pentaho D.I., invece, si basa sull'intuitività dei comandi, attraverso un approccio di programmazione grafico con un ambiente "drag and drop" .

L'obiettivo di questo Elaborato è la generazione di uno schema integrato a partire da un set di sorgenti dati eterogenee, analizzando le differenze di utilizzo e le difficoltà sorte per la produzione di uno schema integrato, mostrano i pregi e difetti di questi due Software.

MOMIS

MOMIS è un sistema di Integrazione di Dati Open Source che sfruttando la semantica presente nelle sorgenti informative è in grado di integrare sorgenti dati eterogenee (strutturate e semi-strutturate) e distribuite. [1]

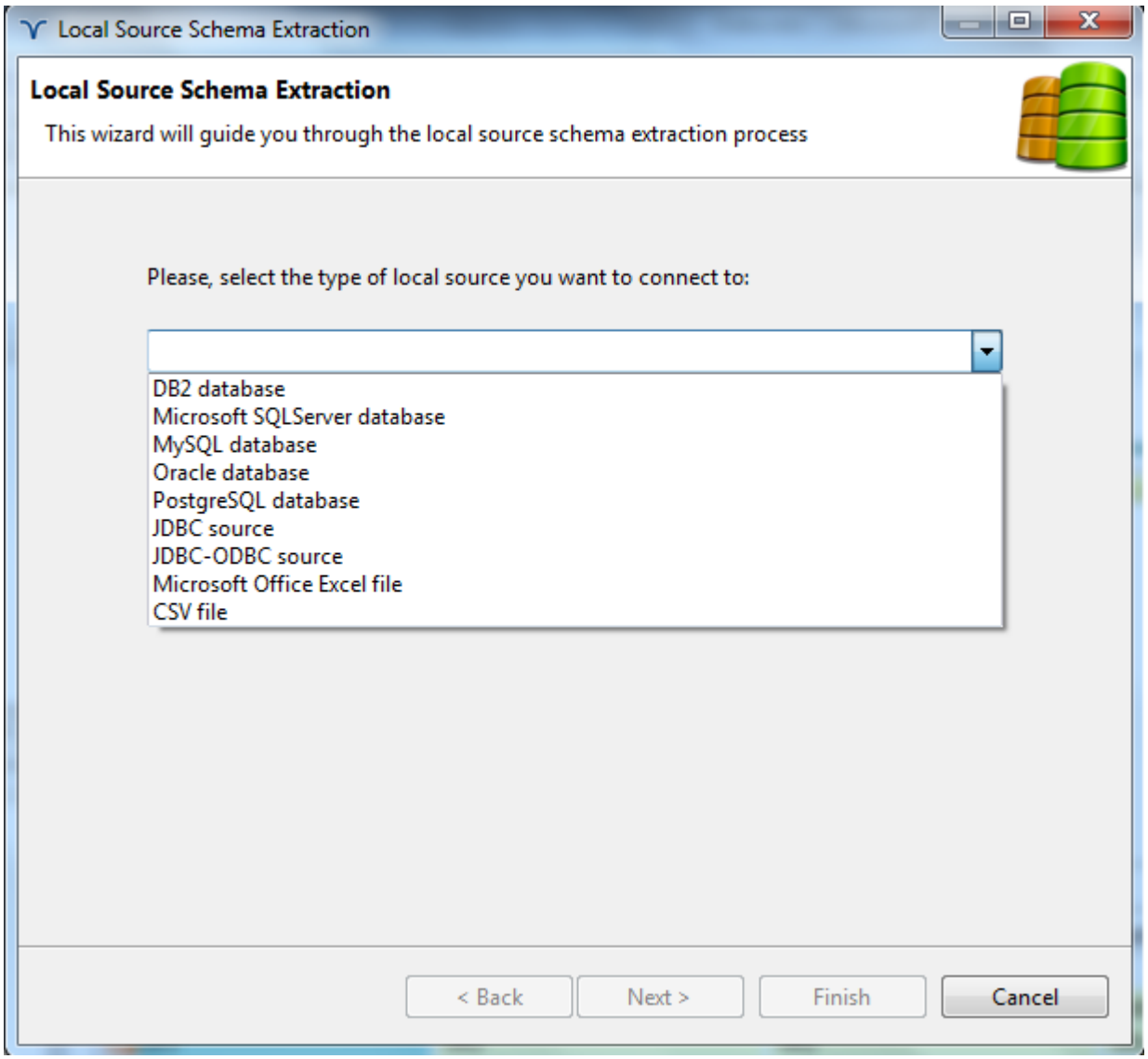
Il software è giunto alla release 1.2 ed è scaricabile previa registrazione gratuita dal sito web: http://www.datariver.it/; sono disponibili varie versioni del sw: con jre (Java Runtime Environment) incluse (nel caso sul nosto computer non fosse installata ne una jre o una jdk), per Windows (32/64bit) Mac (MacOsX 32/64bit) o Linux (32/64bit).

Sullo stesso sito sono disponibili i codici sorgenti, un comodo VideoTutorial suddiviso in 8 parti e un Tutorial Testuale.

Il programma una volta estratto è subito funzionante, alla prima esecuzione ci verrà chiesto di accettare le condizioni d'uso e di scegliere in quale cartella posizionare il "Workspace" ; E' inoltre presente un piccolo progetto dimostrativo (introdotto nella versione 1.2 di MOMIS) per poter subito capire e testare le funzionalità di questo software di data integration.

L'interfaccia davanti alla quale ci troviamo ha i comandi essenziali per effettuare un'integrazione: possiamo procedere con la creazione di un nuovo progetto e l'aggiunta di sorgenti di dati. Questo software è capace di effettuare l'integrazione di dati a partire da **sorgenti** quali:

- DB2 database
- Microsoft SQLServer Database
- MySQL Database
- Oracle Database
- PostgreSQL Database
- Sorgenti JDBC
- Sorgenti JDBC-ODBC
- File Excel
- File CSV
- Altre risorse tramite WEB



Una volta effettuata la connessione alla risorsa, ci verrà chiesto di selezionare le tabelle/attributi/viste da importare.

Terminata questa prima procedura, il programma ci darà la possibilità di aggiungere altre risorse, in caso contrario ci ricorderà di creare un nuovo **Schema Globale.**

Il processo di integrazione si basa su quattro passaggi guidati:

1. Edit Local Source
2. Sources Annotation
3. Semantic Relationship
4. Mapping Refinement

Il **primo** passo è quello dedicato alla scelta delle risorse da integrare : cliccando con il tasto destro del mouse sulla risorsa presente nel "Source Explorer" e selezionando il comando "add selected source to the global schema", è possibile aggiungere allo Schema Globale tutte le risorse a noi necessarie per l'integrazione di dati.

Il **secondo** step è quello caratteristico di MOMIS: le Annotazioni.
Il processo di Annotazione serve per associare le classi e i nomi degli attributi a uno o più significati con un riferimento lessicale comune. In Momis questo processo è basato sul DataBase *WordNet* [1] che connette i termini per mezzo delle Relazioni Semantiche in esso presenti. L'utente quindi può selezionare una forma base del termine oppure si può affidare all' Annotazione automatica attraverso l'ampia rete di relazioni semantiche presenti fra i termini. Le relazioni semantiche in questione sono quelle di "Iponimia" (hyponym) e "Ipernimia" (hypernym) , che rispettivamente legano un termine ad un altro con una relazione di "Specializza" (es. telephone is hyponym of telecommunication) e "Generalizza" (telecommunication is hypernym of telephone). E' inoltre presente una comoda funzione "Hypernym Chart" , che ci permette di vedere graficamente i legami presenti fra i termini base presenti nel DataBase di WordNet. MOMIS ci offre anche la possibilità di importare le Annotazioni da altri Global Schema sviluppati in precedenza e di aggiungere nuovi termini al DataBase di WordNet, questo strumento è chiamato WordNet Extender.

Il passo successivo è l'**estrazione di Relazioni Semantiche** : partendo dalle relazioni presenti negli schemi locali (che possono essere Strutturali=>derivanti dalla struttura dello schema o Lessicali=> derivanti dalle annotazioni introdotte nel passo precedente, ottenute basandoci su WordNet) MOMIS genera in automatico relazioni semantiche inter/intra-schema, come per esempio quella di "sinonimia" (SYN => synonyms), "più specifico/ meno specifico di" (NT/BT=>narrower / broader terms) o semplicemente "correlato a" (RT => related terms) che possono essere modificate e corrette manualmente secondo le nostre esigenze. Inoltre tramite l'ODB Tools (description logic engine) integrato , MOMIS riesce a creare le "Inferred Relationship" (relazioni dedotte) derivanti da logiche descrittive di equivalenza .

---

[1] *WordNet: è un DataBase lessicale in lingua inglese sviluppato da George A. Miller (Docente di Psicologia dell'Università di Princeton) a partire dagli anni '80, che associa un LEMMA (forma base di un termine) con uno o più SYNSET (significato di Lemma) che a sua volta è collegato con un GLOSS (che è una breve definizione del Synset). (http://wordnet.princeton.edu/). [2] [3]*

Infine , l'ultimo passo per completare la nostra integrazione di dati è quello del **Data Fusion Process** , quindi della generazione di "Cluster" (raggruppamenti) e del Mapping degli attributi appartenenti alle risorse locali in Classi Globali; In questa schermata l'utente può scegliere in che modo completare l'aggregazione al fine di ottenere il risultato organizzato nel modo a lui più gradito ; Si deve scegliere la più appropriata Funzione di Join per ogni classe globale , la (se necessaria) creazione di Funzioni di Trasformazione che ci permettono di ottenere un attributo della classe globale partendo da vari attributi locali o risolvendo i conflitti che si generano nel processo tramite le Funzioni di Risoluzione. L'utilizzatore del software inoltre può decidere di creare manualmente Classi Globali (senza affidarsi "completamente" alle scelte automatiche legate alle relazioni semantiche) e dotarle di Attributi Globali da noi definiti tramite le funzioni sopra citate.

MOMIS integra anche un tool per il test del risultato ottenuto dall' integrazione, possiamo interrogare lo Schema Globale ottenuto usando il **Query Manager Tools**[2]; E' stato inoltre introdotto nelle ultime versioni del software il "Query Plan Viewer" che, ricordandoci dell' approccio GAV (Global As View) su cui il software si basa , ci da la possibilità di visualizzare i tre passaggi con cui il programma esegue le query :

1. (Local Queries) Le query vengono prima eseguite sulle risorse locali in maniera simultanea
2. (Mappings Queries) Il risultato delle prime query è unito /aggregato da MOMIS in un risultato parziale che tiene conto delle nostre relazioni di join impostate nell'aggregazione
3. (Final Query) Una query finale che applica le funzioni di risoluzione da noi impostate e le ultime clausole non ancora risolte

Un'altra funzione presente in MOMIS è quella della "DATA PREVIEW", che ci aiuta a vedere se i passi dell'integrazione sono corretti, in quanto ci fornisce un' anteprima istantanea dei dati presenti negli schemi globali .

PENTAHO Data Integration

Pentaho Data Integration (PDI , prima chiamato Kettle: "Kettle Extraction, Transport, Transformation and Loading Environment") è un modulo del software PENTAHO che consente di estrarre, trasformare e caricare (ETL) i dati da una qualsiasi fonte ; E' un potente strumento con un ambiente di sviluppo grafico ed orientato ai metadati; Grazie a questo strumento si possono incrociare i dati da più fonti , si ha la possibilità di aggiornarli in real time ed effettuare migrazione di dati tra sistemi diversi tramite la creazione di programmi (job) . E' usato per trasferire i dati fra DataBase e Flat Files (file non Strutturati => senza

[2] Un DBMS (DataBase Management System) relazionale dà supporto al Query Manager Tools per la fusione dei risultati parziali in tabelle temporanee: in MOMIS è incluso HyperSQL che permette al programma di essere usato senza nessuna configurazione AD-HOC da parte dell'utente finale e che inoltre mostra tutte le "sub-query" che vengono eseguite per arrivare al risultato.

relazioni strutturali=> per usarli bisogna conoscere la loro formattazione); Ci da inoltre possibilità di salvare i lavori sia in Repository (Deposito dove vengono gestiti i Metadati attraverso tabelle relazionali , ottimo per grandi moli di dati) che Files.

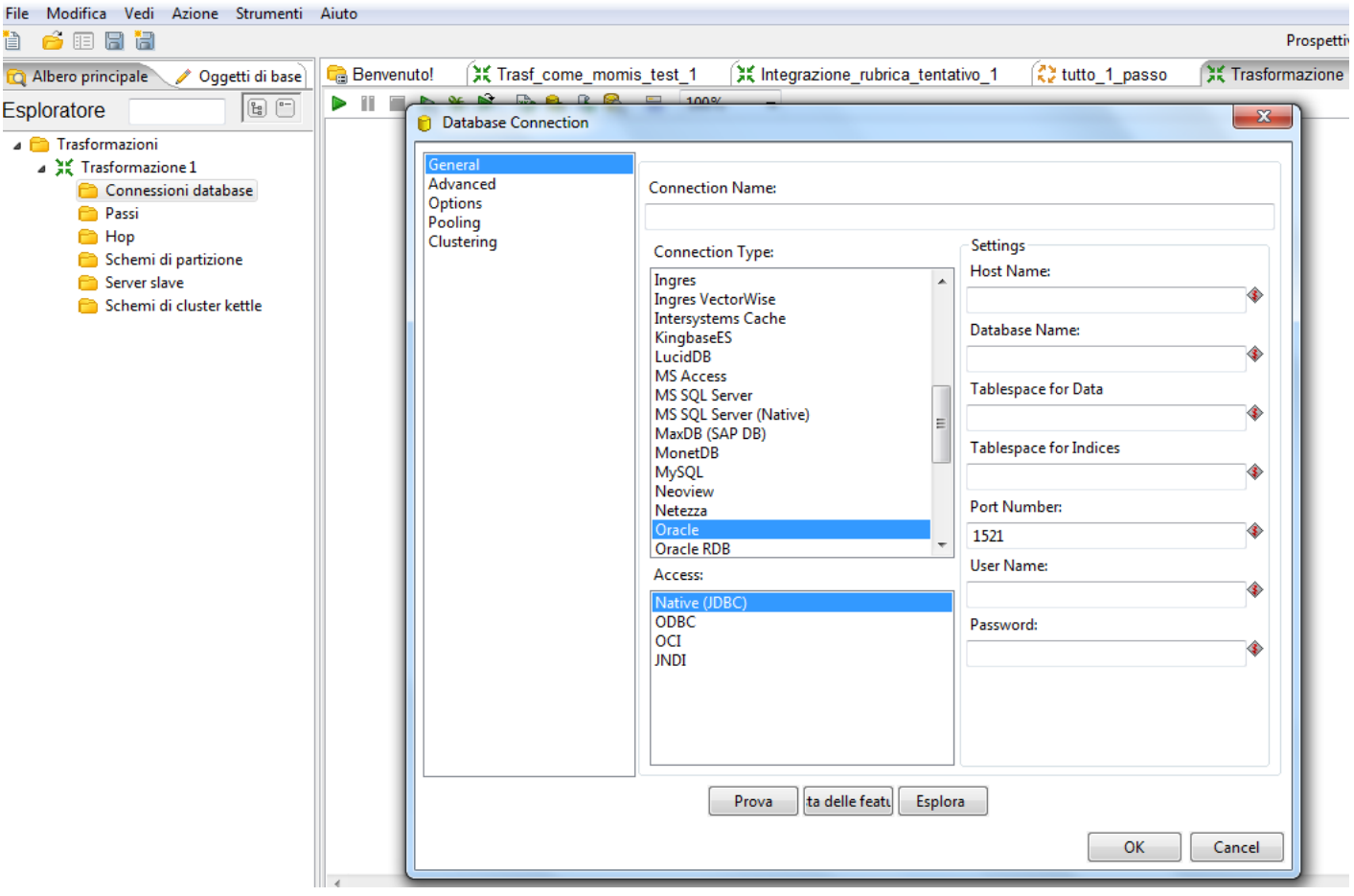
Il software è giunto alla versione 4.3.0 che è possibile ottenere gratuitamente dal sito http://kettle.pentaho.com/ .

Il programma si presenta con i comandi in lingua italiana e con una schermata di benvenuto che ci permette di ottenere più informazioni riguardo al software.

Sul sito di Pentaho D.I. è presente una guida introduttiva, un tutorial troppo semplice per darci un'idea sulle funzionalità del programma e tantissime pagine che spiegano i comandi presenti (ma putroppo non aggiornate all'ultima versione del software).

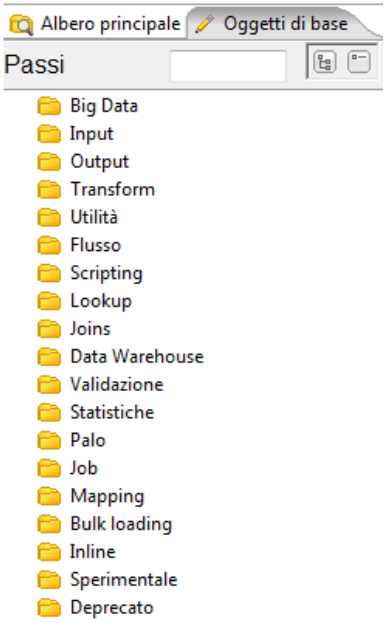
Pentaho Data Integration è formato da quattro componenti: **Spoon** (disegno grafico dei passi dell'ETL), **Pan** (esecuzione da linea di comando delle trasformazioni), **Kitchen** (esecuzione dei job), **Carte** (console per l'esecuzione remota); In questa relazione abbiamo fatto uso solo dell'interfaccia grafica, eseguibile in Windows da "spoon.bat".

Per iniziare il processo di integrazione di dati è necessario prima di tutto creare una nuova Trasformazione (nel nostro caso, altrimenti si può creare un Job) , posizionarci nell'albero principale e creare una nuova connessione a tutte le sorgenti di dati che vogliamo integrare (le possibilità sono moltissime), una volta impostati i collegamenti ai Database di nostro interesse è possibile iniziare il processo che, come vedremo, è grafico ed intuitivo ma non guidato (quindi l'utente che dovrà effettuare il processo dovrà avere bene a mente gli "step" che dovrà eseguire per arrivare al risultato desiderato).

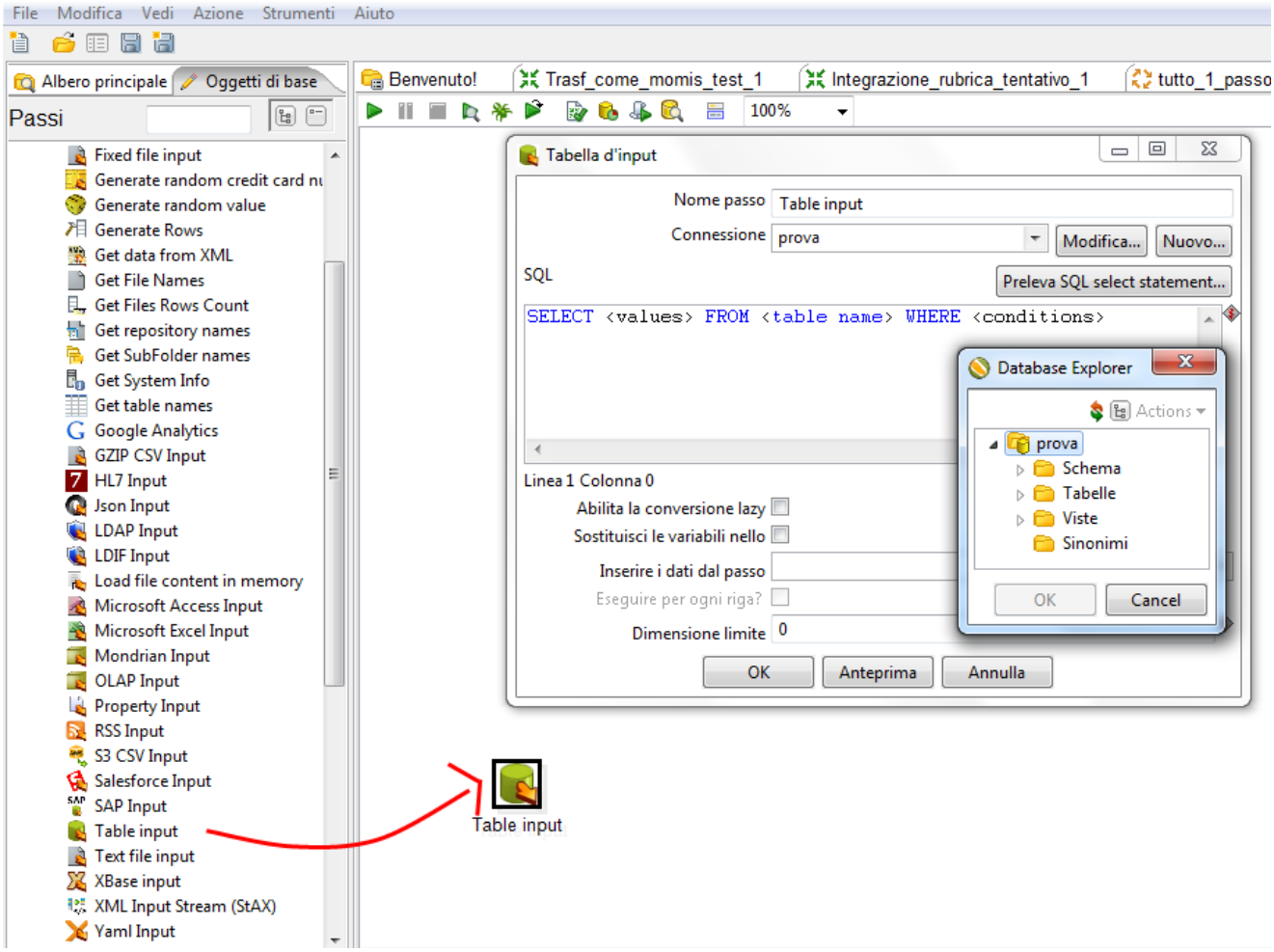


I passi da impostare per la trasformazione sono raggruppati in varie classi:

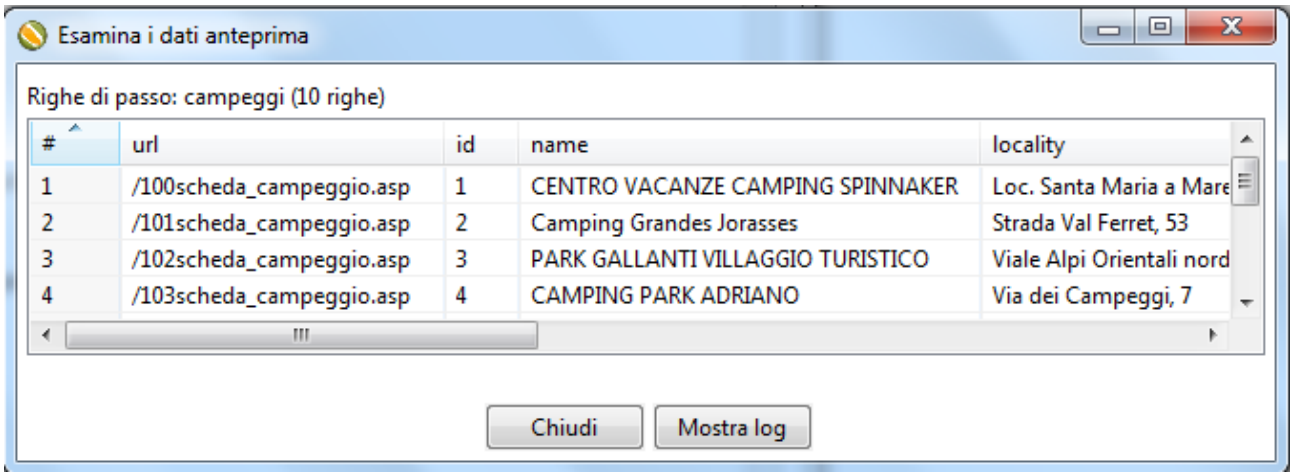
- BIG DATA
- INPUT
- OUTPUT
- TRANSFORM
- UTILITA'
- FLUSSO
- SCRIPTING
- LOOKUP
- JOINS
- DATA WAREHOUSE
- VALIDAZIONE
- STATISTICHE
- Ecc..



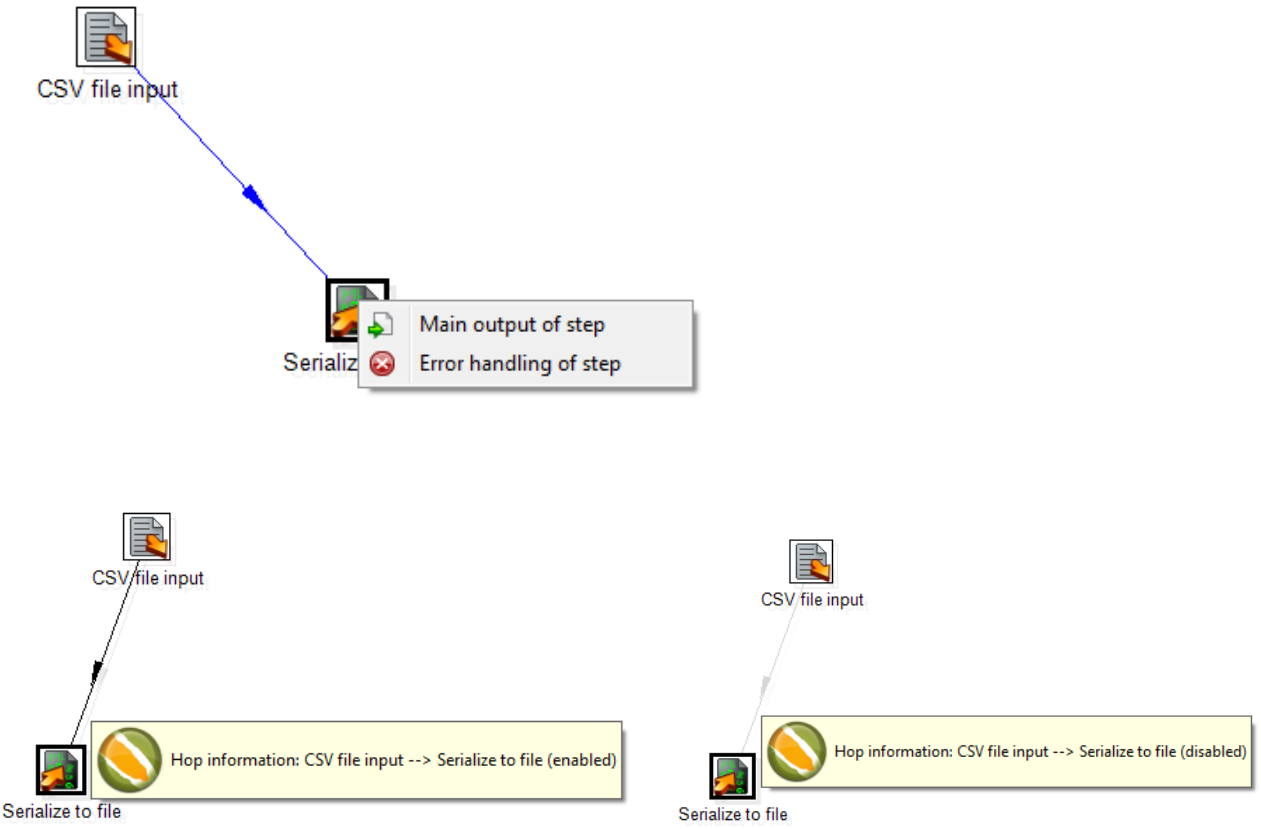
Per prima cosa noi sceglieremo la classe "Input" e trascineremo nello spazio bianco il nostro "table_input", con doppio click sull'oggetto possiamo impostare il nome del passo e da quale connessione proviene , premendo "Preleva SQL select statement" decidiamo quale tabella (o vista) del DB prendere e quali colonne selezionare, in questo menù abbiamo la possibilità di scrivere codice SQL ed usare funzioni provenienti da quel linguaggio (es. coalesce , order by , group by, ecc..)



Pentaho ci offre la possibilità di avere un' anteprima dei dati in Input , potendo anche selezionare quante righe visualizzare.



I passi devono essere collegati fra di loro tramite HOP, che si creano trascinando lo step di partenza sullo step di arrivo tenendo premuto il tasto "SHIFT", selezionando poi che tipo di collegamento si tratta (output principale o output d'errore); E' possibile disabilitare il collegamento con un semplice click sul medesimo e selezionandolo un'altra volta si ri-attiverà.

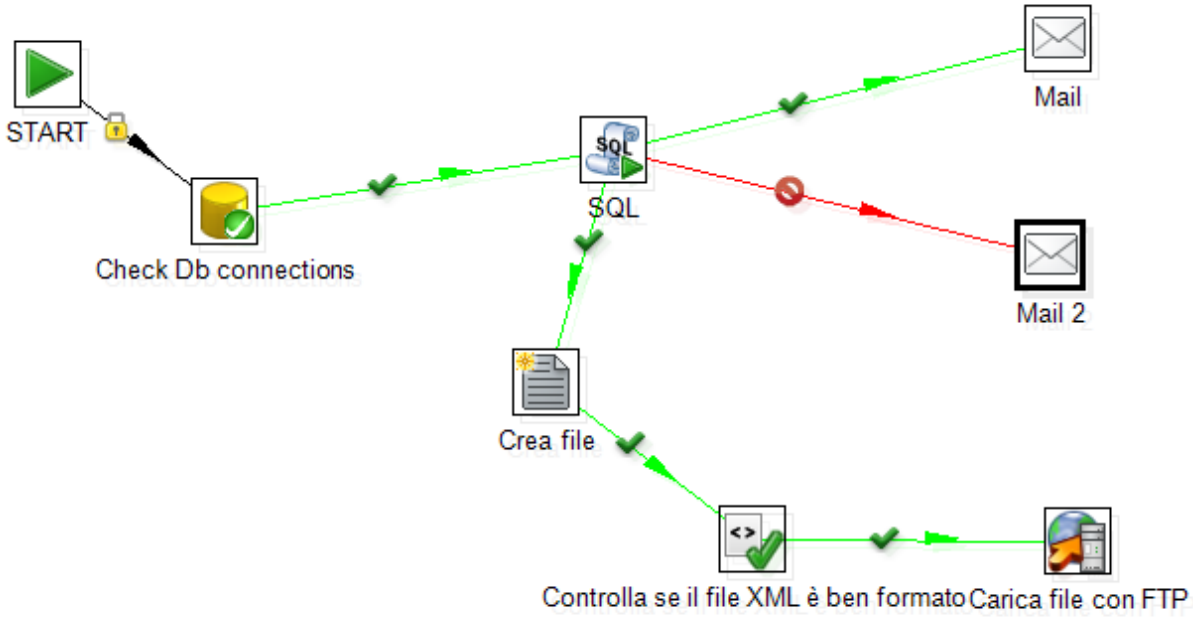


Dopo che i vari passi modificano/aggregano i dati provenienti dall'input possiamo portarli in output su tabelle, file di testo, fogli di Excel, ecc..

Con Pentaho D.I. possiamo utilizzare script in JAVA da affiancare alle funzioni "native" del sw per avere determinati comportamenti non altrimenti ottenibili.

Possiamo notare come a differenza di MOMIS, l'utente che vuole utilizzare Pentaho necessita di un' ottima conoscenza di Basi di Dati , di SQL e dello studio degli oggetti che esso mette a disposizione , in quanto non è guidato.

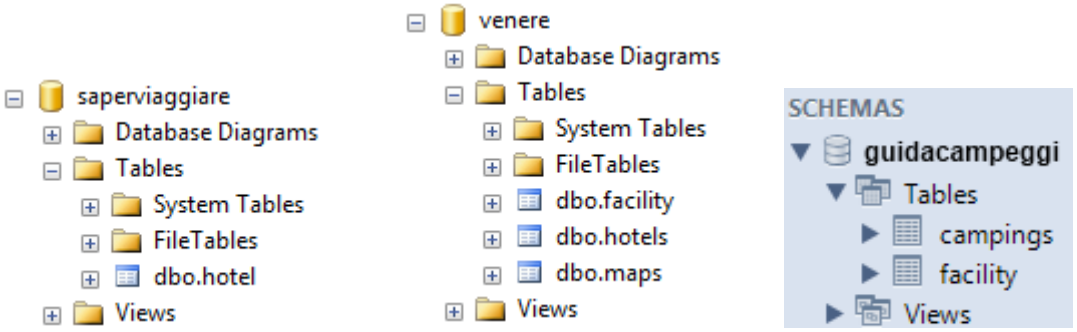
Oltre che le trasformazioni (che ci permettono di "trasportare/trasformare" dati da una sorgente ad un'altra) , Pentaho Data Integration mette a disposizione un altro strumento chiamato JOB: esso è uno strumento che ci permette di "accorpare" e coordinare fra loro differenti trasformazioni o attività (es. invio o la scrittura di un File , di una eMail, trasferimento di alcuni dati, il controllo su un DataBase, ecc..).



E' da notare che se si vuole testare l'integrazione di dati effettuata, si necessita di un DBMS esterno, in quanto Pentaho D.I. non fornisce nessun tools per la creazione di query .

TEST N°1

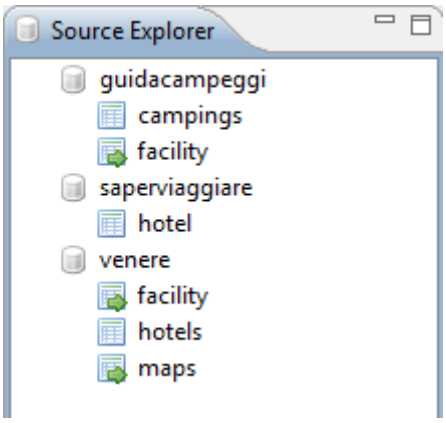
Il test primo test da me effettuato è basato sulle stesse sorgenti dati del progetto esempio precaricato in Momis, con la differenza che abbiamo effettuato l'integrazione avendo i dati posizionati su diverse sorgenti (nel mio caso due caricate su Microsoft SQLServer 2012 e una su MYSQL, rispettivamente "SaperViaggiare" e "Venere" su MSSQL Server e "GuidaCampeggi" su MySQL) . L'obiettivo dell' integrazione di dati è ottenere due tabelle: la prima che raggruppi le informazioni relative a tutti i Campeggi ed Hotel (chiamata STRUCTURE) , e l'altra che contenga tutti i servizi offerti dalle strutture (Facility).



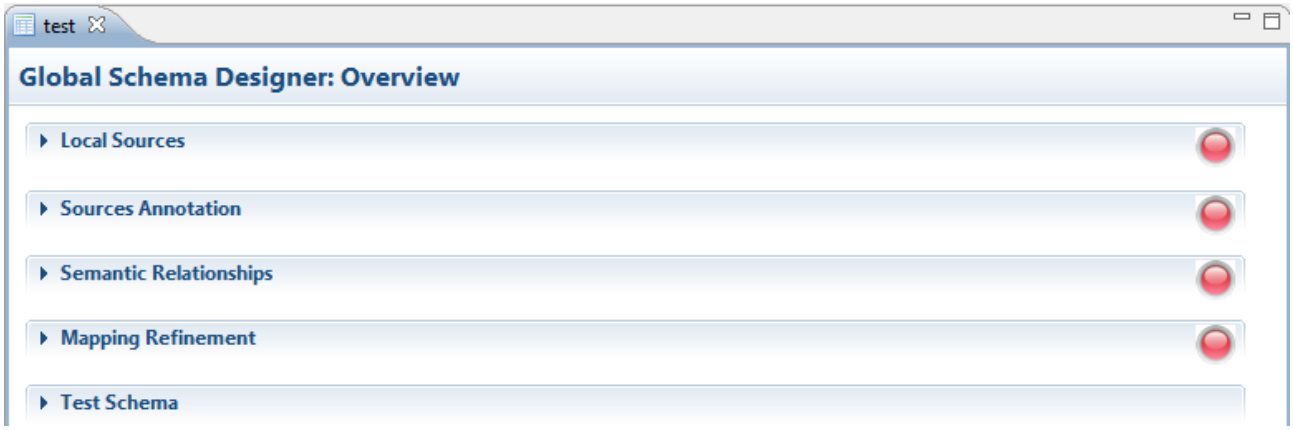
MOMIS

Effettuando l'integrazione con MOMIS notiamo subito che, dall'apertura del programma , tutti i passaggi necessari per l'integrazione di dati sono guidati, come per esempio l'aggiunta di sorgenti : basta selezionare da che tipo di risorsa ricevere i dati, inserirgli un nome di riferimento, specificare l'indirizzo , inserire i parametri per il log-in nel server e scegliere il DataBase di nostro interesse.

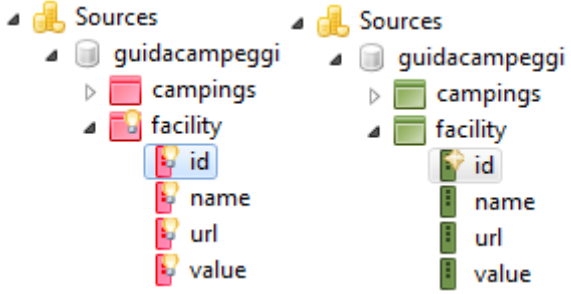
Una volta aggiunte tutte le risorse, creeremo un nuovo **Schema Globale** e quindi inizieremo il processo per l'integrazione di queste sorgenti eterogenee di dati.



Ci troviamo davanti alla schermata "Overview" , che ci mostra a che punto dell'integrazione siamo arrivati : è presente un segnalino rosso/verde accando ad ogni Passo che indica lo stato di completamento.

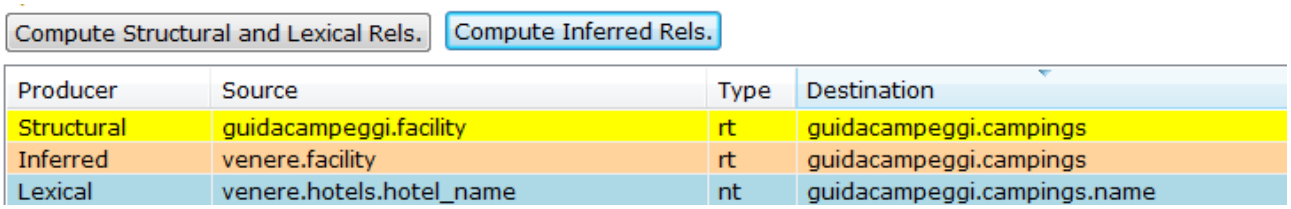


La prima azione da eseguire è quella di aggiungere le risorse locali allo Schema Globale, noi aggiungeremo "GuidaCampeggi", "SaperViaggiare" e "Venere", dopo di che è possibile passare allo Step successivo: le Annotazioni. Momis a questo punto ci ricorderà di salvare il progetto.



Il processo di Annotazione delle risorse locali presenti nello schema è il passo fondamentale per l'integrazione dei dati: è possibile effettuare l'annotazione automatica , per poi raffinare il risultato a mano (per gli abbinamenti che nel nostro caso potrebbero essere errati oppure per i termini che risultano essere sconosciuti al programma); Come spiegato nella presentazione del programma , MOMIS si basa sul WordNet : un DataBase lessicale in Lingua Inglese che contiene al suo interno moltissime relazioni fra i termini (relazioni di Iponimia, Ipernimia). Viene inoltre usato WordNetExtender per aggiungere nuovi vocaboli e relazioni al DB. Bisogna dare molta importanza a questo passo, in quanto una buona annotazione dei termini ci permette di facilitare il lavoro delle prossime fasi, in particolare per la generazione di Cluster. Possiamo notare che con l'annotazione automatica, il software ha inserito come significato della colonna "address" un Synset del "Ramo Informatico", quindi noi andremo a modificarlo invece con il significato di: "The place where a person or organization can be found or communicated with", che nel nostro contesto risulta essere più appropriato.

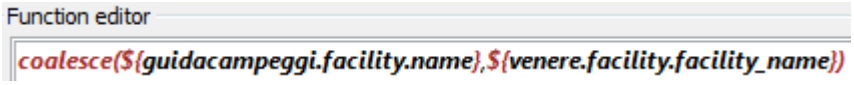
Ora dobbiamo definire le Relazioni Semantiche presenti nelle sorgenti: MOMIS ci aiuta ancora una volta, proponendoci la creazione di una serie di Relazioni Lessicali e Strutturali automatiche. E' inoltre possibile aggiungere delle relazioni manualmente (tramite pulsante "ADD"). Infine con il pulsante "Compute Inferred Relations" il programma calcola le relazioni dedotte da quelle già inserite (tramite l' ODBTool) e le aggiunge a quelle presenti .



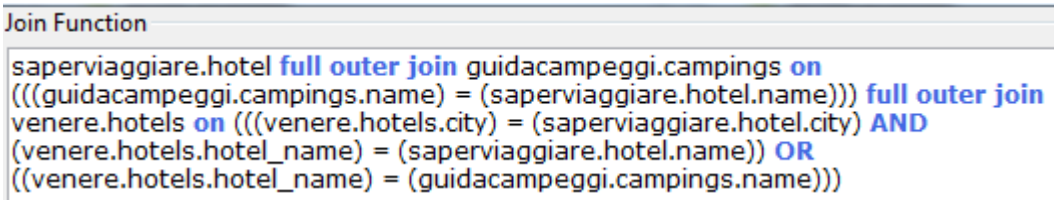
L'ultima azione da effettuare prima di ottenere l'integrazione è quella della scelta delle "impostazioni di Clustering": Momis tramite degli sliders ci aiuta nella creazione dei Cluster dei dati. Dopo di chè sarà compito dell'utente "riorganizzare" secondo le neccessità di aggregazione gli attributi locali in attributi globali. Nel nostro caso vogliamo ottenere due Classi Globali:

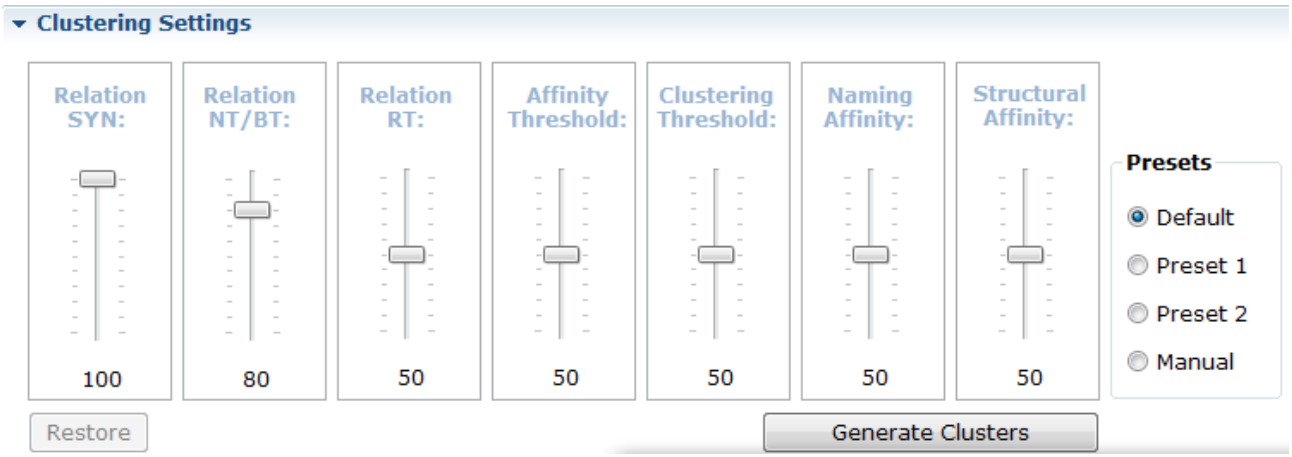
- "Facility" che conterrà l'elenco dei servizi di tutte le Strutture;
- "Structure" che conterrà l'elenco di tutti gli Hotel e Campeggi;

Iniziamo con il definire Facility: nella sottosezione "Mapping Refinement" possiamo eliminare le classi create in automatico (nel caso Momis le avesse create non conformi con le nostre aspettative) e crearne di nuove (tasto dx su Global Source=>"Add global Class") che poi popoleremo con nuovi Attributi Globali. Gli Attributi Globali li possiamo ottenere semplicemente per "aggiunta" oppure tramite la trasformazione/ unione / risoluzione degli attributi locali . Nel nostro test, per esempio, abbiamo usato come funzione di Risoluzione per il campo "facility_name" in "Facility" la funzione "Coalesce" , in quanto dal join tra le due classi contenute in "GuidaCampeggi" e "Venere" ci risulterà un campo "NULL" e l'altro contenente il Valore. Abbiamo definito anche una funzione di Trasformazione per il nuovo campo "Rating" di Hotel che sarà ottenuto dall'estrazione della cifra del campo "rating" e "user rating" calcolandone la media.

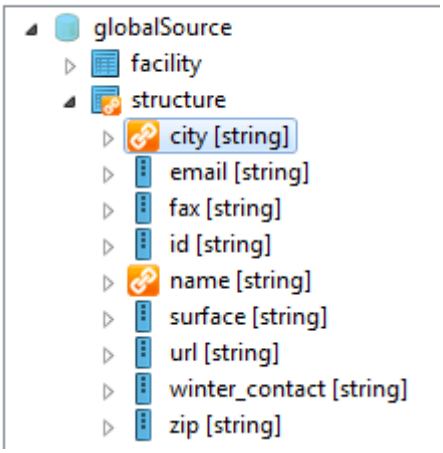


Per ottenere l'aggregazione dei Dati dobbiamo selezionare quali sono gli attributi su cui verrano effettuati i Join dal Programma, e nel caso la funzione che viene generata in automatico non fosse a noi utile, possiamo modificarla manualmente andando nell'opzione "Edit Join Function" sulla Classe Globale.





Global Source



Mapping Table: structure

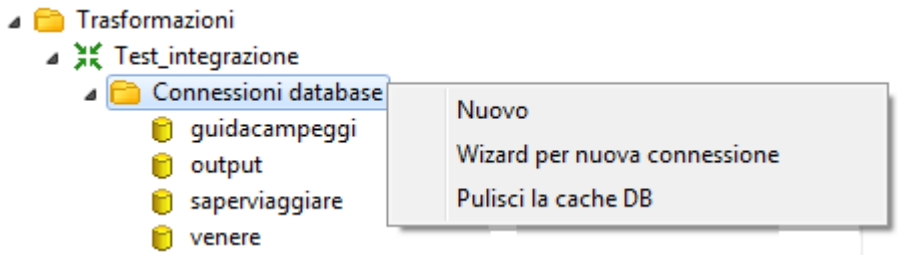
structure(globalSource)	campings(guidacampeggi)	hotel(saperviaggiare)	hotels(venere)
city	city	city	city
email	email	email	
fax	fax		
id	id		id
name	name	name	hotel_name
surface	surface		
url	url	url	url

Ora che abbiamo finito la creazione delle nuove Classi Globali, ottenute dall'integrazione dei dati provenienti da varie Sorgenti, è possibile testare il nostro risultato tramite l'opzione "Launch Query Manager" presente nella sottosezione "Test Schema" della schermata "Overview" . Da questa sezione del programma, oltre che a testare il risultato, è possibile vedere come in realtà esso lavori per risolvere le query da noi inserite: prima verranno effettuate delle query su ogni risorsa locale, dopo il risultato verrà semi-aggregato in un risultato parziale e solo alla fine MOMIS risolverà le ultime clausole per arrivare al risultato finale.

PENTAHO Data Integration

Eseguendo la stessa integrazione di dati eseguita anche con MOMIS partendo dalle stesse sorgenti, possiamo notare che in questo Software ogni azione da eseguire deve essere esplicitamente scelta e configurata dall'utente.

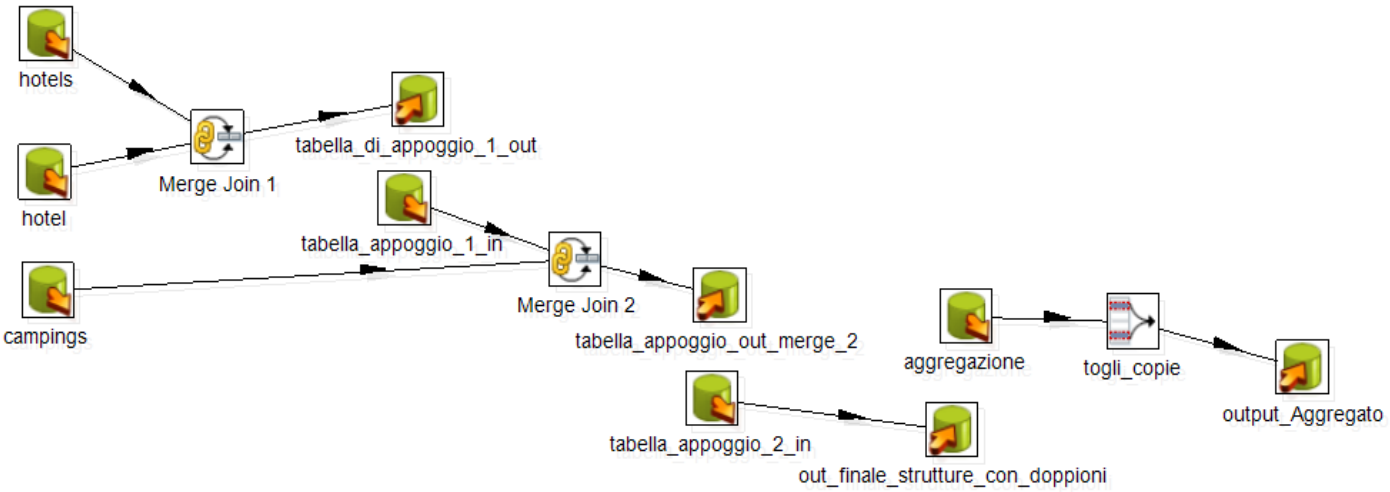
Poiché il nostro obiettivo è quello di ottenere un'integrazione di dati eterogenei, creiamo una nuova Trasformazione e subito aggiungiamo le connessioni alle sorgenti di nostro interesse: andremo in "Albero Principale" , tasto destro su "Connessioni" => "Nuovo" ed impostiamo i campi richiesti ; una volta finita questa semplice procedura la ripeteremo per ogni sorgente .



Iniziamo ad aggiungere al nosto schema tutti gli elementi "Input" per ogni tabella con cui vogliamo lavorare, ricordandoci di configurare ogni elemento dal proprio menù che viene richiamato tramite doppio click sull' oggetto.

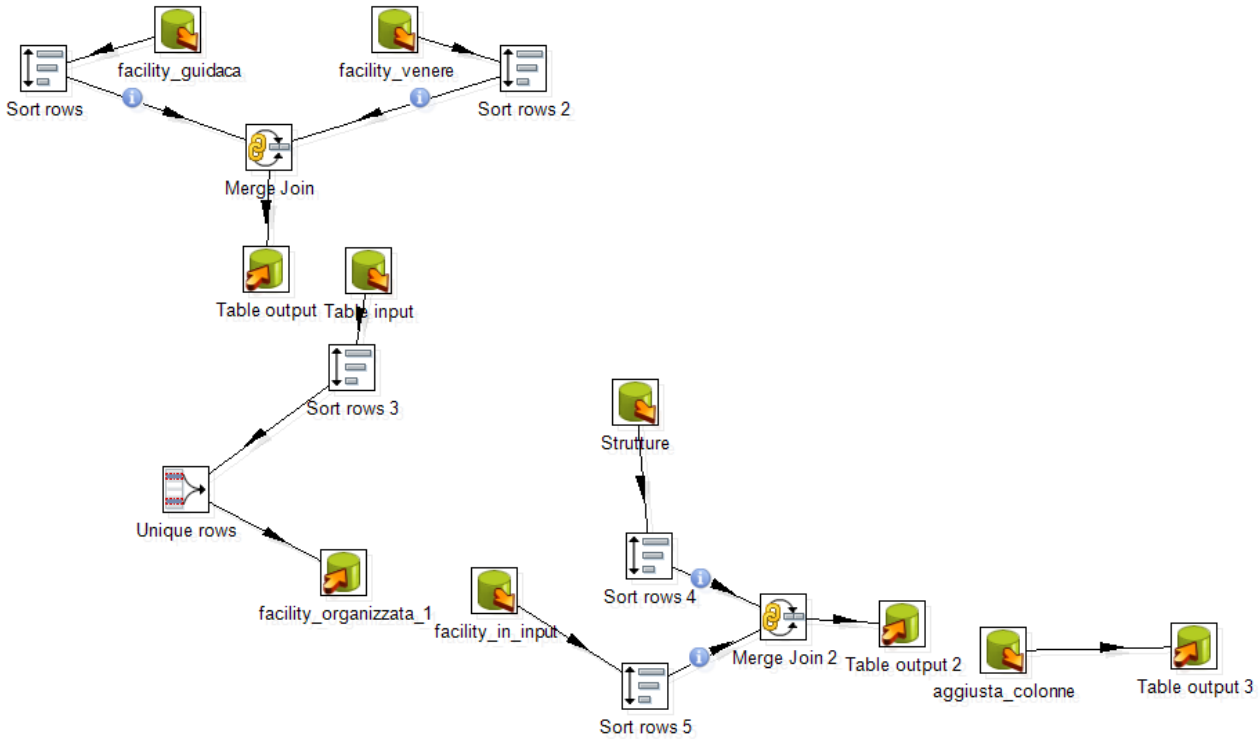
In questo test, l'integrazione di dati è stata ottenuta mediante l'uso dell'elemento "Merge Join" che prende due flussi di dati in input ed esegue un Join fra questi quattro tipi: Inner , Left Outer , Right Outer e Full Outer ; Per l'integrazione dei dati presenti nelle tabelle Hotel (contenuta in SaperViaggiare) e Hotels (in Venere) abbiamo usato un Full Outer Join effettuato sui campi "Città" e "Nome", in modo da avere in output una tabella con tutti gli Hotel ripetuti una sola volta.

Ora vogliamo Integrare insieme agli Hotel anche i Campeggi, così da avere tutti i dati in un'unica tabella che chiameremo "Structure": anche qui, per nostra sicurezza, abbiamo effettuato un Full Outer Join (anche se eravamo abbastanza sicuri che nessun Hotel fosse presente nella tabella Campeggi) .



L'altra parte dell'integrazione effettuata riguarda le due tabelle Facility presenti nei Database "SaperViaggiare" e "Venere". Queste tabelle contengono un elenco di tutti i servizi offerti dalle

varie Strutture avendo come riferimento il loro URL , quindi per avere un'aggregazione delle informazioni strutturata in maniera "più comoda / utile " abbiamo effettuato un "Right Outer Join" fra "Structure" appena creata e l'unione delle due tabelle "Facility", selezionando poi i campi "Città" , "Nome_struttura" , "Url" e "Facility_name" .



Ottengo così la possibilità di effettuare una query del tipo:

*select * from "tabella_ottenuta" where nome_struttura = 'Di_nostro_interesse';*

che ci informa su tutti i servizi della struttura di nostro interesse;

oppure :

*select * from "tabella_ottenuta" where Facility_name = 'servizio_a_noi_necessario';*

che ci mostra quali sono le strutture con il servizio da noi voluto (es. dog_accepted, oppure Restaurant).

*Select * from "tabella_ottenuta" where Facility_name='servizio' and City='Firenze'*

Se necessitiamo di sapere quale struttura di una determinata città offre un determinato servizio.

```
select * from prova_rendere_facility_utile_p4_colonne_selezionate
        where name='CENTRO VACANZE CAMPING SPINNAKER';
```

Results | Messages

name	city	url	Facility_name
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Washbasins
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Basket
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Bathroom
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Bathroom for disabled
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Dogs accepted
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Laundry room
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Market
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Private Parking
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Restaurant
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Shops
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Shower
CENTRO VACANZE CAMPING SPINNAKER	Fermo	/100scheda_campeggio.asp	Bar

```
select * from prova_rendere_facility_utile_p4_colonne_selezionate
        where city='Firenze' and Facility_name = 'Sauna';
```

esults | Messages

name	city	url	Facility_name
Villa La Vedetta	Firenze	/firenze/7hotel_villa_la_vedetta.html	Sauna
Villa Medici	Firenze	/firenze/29hotel_villa_medici.html	Sauna
adler Cavalieri	Firenze	/firenze/5hotel_adler_cavalieri.html	Sauna

La difficoltà principale che si incontra a lavorare con questo software è l'estrema libertà (questa volta intesa in senso negativo, cioè l'assenza di linee guida per effettuare l'integrazione) che obbliga un nuovo utente, per la prima volta alle prese con Pentaho Data Integration, a ricorrere all'aiuto non sempre "preciso" del "Pentaho Infocenter[3]" (in quanto non aggiornato all'ultima release 4.3.0). Oltretutto bisogna avere già da principio le idee chiare su che tipo di integrazione voler eseguire e con che criterio procedere per ottenere tale risultato.

Il tempo necessario per creare il processo di integrazione risulta essere maggiore rispetto a quello impiegato usando MOMIS: questo è il costo che l'utente deve pagare per avere a disposizione molti più strumenti per effettuare trasformazioni "non standard" (es. ottenere dati presenti su file di testo, i nomi di file presenti in una cartella, email , info di sistema, ecc..), senza utilizzare altri software di supporto.

[3] http://infocenter.pentaho.com/help/index.jsp

TEST N°2

Il secondo test di Integrazione è stato effettuato fra un DataBase e due File Parametri (nel nostro caso file Excel) strutturati in questo modo:

Il DB è chiamato SRCDEMO e contiene questa tabella :

SRC_CUSTOMER			
CUST_ID	NUMERIC (10)	<PK>	NOT NULL
DEAR	NUMERIC (1)		NULL
LAST_NAME	VARCHAR (50)		NULL
FIRST_NAME	VARCHAR (50)		NULL
ADDRESS	VARCHAR (100)		NULL
CITY_ID	NUMERIC (10)		NULL
PHONE	VARCHAR (50)		NULL
AGE	NUMERIC (3)		NULL
SALES_PERS_ID	NUMERIC (10)		NULL

I file parametri invece sono così strutturati:

SRC_SALES_PERSON			
SALES_PERSON_ID	NUMERIC (10)	<PK>	NOT NULL
FIRST_NAME	VARCHAR (50)		NULL
LAST_NAME	VARCHAR (50)		NULL
HIRE_DATE	DATE		NULL

SRC_AGE_GROUP			
AGE_MIN	NUMERIC (3)	<PK>	NOT NULL
AGE_MAX	NUMERIC (3)	<PK>	NOT NULL
AGE_RANGE	VARCHAR (50)		NULL

L'esempio è stato in parte preso dalla pubblicazione **"Oracle Data Integrator : Getting Started with an ETL Project".** [4] [5]

L'obiettivo di questa integrazione è: partendo dai dati contenuti in src_customer (tabella che contiene i clienti) e da due file che contengono i dati dell'età (src_age_group) e i dati dei venditori (src_sales_person) , popolare una tabella di destinazione (trg_customer) seguendo alcune logiche:

- Join fra venditori e clienti deve essere fatto tramite *src_customer.sales_pers_id= src_sales_person.sales_person_id*;
- L'età dei clienti deve essere compresa fra quella minima e quella massima presente nel file src_age_group;
- Il campo DEAR di destinazione deve essere : "MR" se *src_customer.dear=0*, "MRS" se questo campo è 1 , "MS" altrimenti;
- Cust_name deve essere la concatenazione del nome e del cognome del cliente, in modo da averli in un unico campo;
- Sales_pers deve essere la concatenazione del nome e del cognome del venditore, in modo da averli in un unico campo;
- Gli altri campi vengono semplicemente "ricopiati" dalla struttura d'origine;

Sono presenti inoltre due vincoli aggiuntivi:

1. L'età dei clienti deve essere maggiore di 21 anni
2. Il codice identificativo della città (city_id) deve esistere in un'altra tabella trg_city (locata nel DataBase TRGDEMO).

Le tabelle ora in prova sono popolate nel seguente modo:

SRC_CUSTOMER

Results | Messages

	Custid	Dear	Last_name	First_name	Address	City_id	Phone	Age	Sales_pers_id
1	1	0	SANTESE	MARCO	VIALE PANORAMA	73046	833510048	21	1
2	2	0	PALLINO	PINCO	VIALE VATTELA	41121	599999	30	2
3	3	0	DE CESARI	MARIO	VIA MARE	41126	895555	38	4
4	4	0	FERRARO	FILIPPO	VIA CALVARIO	73010	833518888	22	3
5	5	0	COGNOM	NOM	VIA STRANA	73100	832510858	40	5
6	6	0	MARRA	VALERIO	VIALE MATTINO	73046	9999999	99	6
7	7	1	ANGELE	SIMONETTA	PIAZZA UMBERTO	73046	481895	23	7
8	8	2	de nicola	laura	PIAZZA rosa	73040	4814895	40	7
9	9	2	CARPA	SIRIA	VIA ALBERI	6123	2313121	18	8
10	10	1	SCHIRINZI	PAOLO	VIA MANO	41126	23133	40	7
11	11	0	CAIRA	ROBERTO	VIA PERUGIA	6123	23133	25	7

TRG_CITY

Results | Messages

	city_id	region_id	city	population
1	4111	3	firenze	150000
2	41121	1	modena_a	160000
3	41126	1	modena_b	140000
4	73010	3	casarano	24000
5	73046	3	matino	14000
6	73100	3	lecce	94000

NB: abbiamo dovuto inserire una linea contenente i nomi delle colonne (come intestazione).

SRC_SALES_PERSON [modalità compatibilità]

	A	B	C	D
1	SALES_PERSON_ID	FIRST_NAME	LAST_NAME	HIRE_DATE
2	1	VENDE	TUTTO	17/09/2012
3	2	marco	gibboni	15/09/2012
4	3	antonio	facaldo	05/10/2011
5	4	lorenzo	carlo	01/01/2012
6	5	GERMANO	STRANO	05/05/2012
7	6	MATTEO	DE MATTEIS	08/03/2011
8	7	FRANCESCO	ANTONACI	08/08/2008
9	8	MINO	RAJOLA	22/04/2005
10	9	ALBERTO	RIMEDIO	08/10/2006

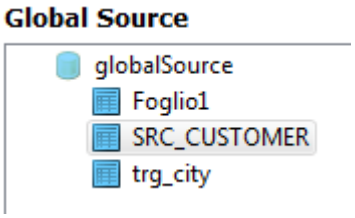
SRC_AGE_GROUP [modalità compatibilità]

	A	B	C
1	AGE_MIN	AGE_MAX	AGE_RANGE
2	30	49	F

MOMIS

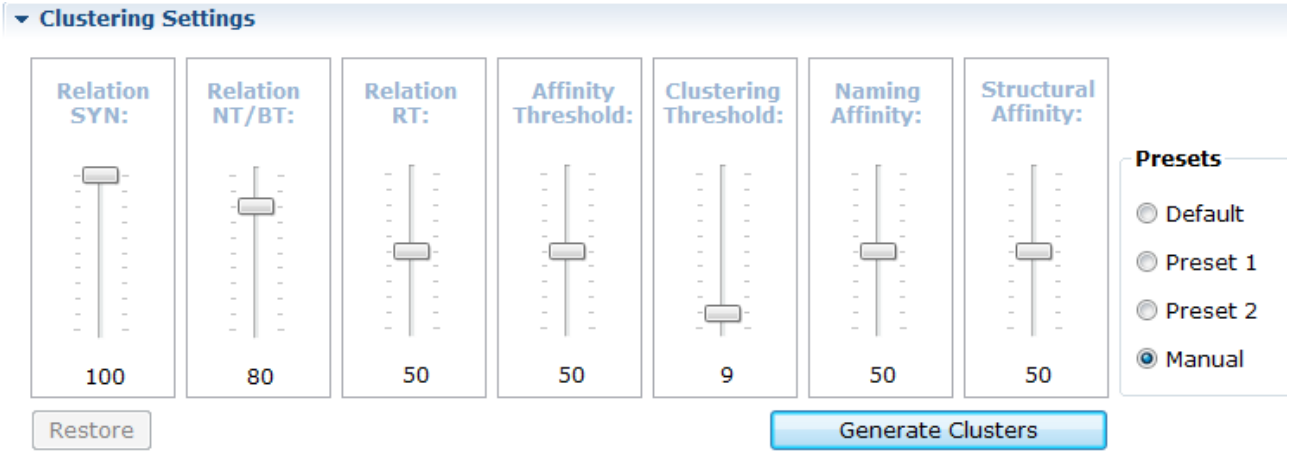
Con MOMIS non ho potuto raggiungere l'obiettivo prefissato in questo Test, in quanto MOMIS non permette di Materializzare le viste, quindi non sono riuscito ad inserire i risultati dell'integrazione nella tabella di destinazione.

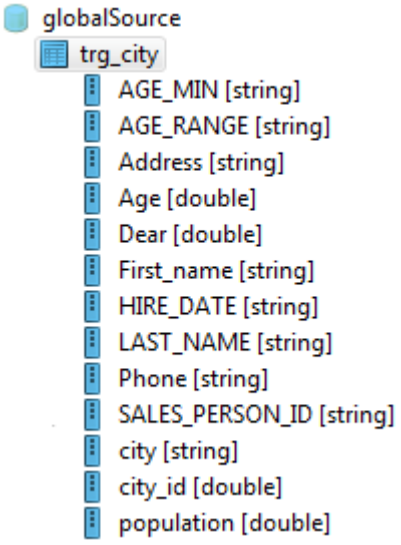
Dopo aver effettuato l'aggiunta delle risorse ad un nuovo progetto, dopo aver creato un nuovo Schema Globale, dopo aver aggiunto le Annotazioni ed aver definito le Relazioni fra le varie classi, ci troviamo nella schermata dove dobbiamo impostare i parametri di aggregazione: nel nostro esempio abbiamo provato a generare i cluster con l'imposazione di default, ma il risultato ottenuto non è soddisfacente



Abbiamo ottenuto tre classi globali.

Ho provato inoltre ad impostare manualmente i valori degli sliders delle “Cluster Settings”, portando al valore 9 il campo “Clustering Threshold”.





In questo modo ottengo solo una classe globale: trg_city.

Mapping Table: trg_city

trg_city(globalSource)	Foglio1(src_age_group)	Foglio1(src_sales_person)	SRC_CUSTOMER(src)	trg_city(trg)
AGE_MIN	AGE_MIN			
AGE_RANGE	AGE_RANGE			
Address			Address	
Age	AGE_MAX		Age	
Dear			Dear	
First_name		FIRST_NAME	First_name	
HIRE_DATE		HIRE_DATE		
LAST_NAME		LAST_NAME	Last_name	
Phone			Phone	
SALES_PERSON_ID		SALES_PERSON_ID	Sales_pers_id , Cu...	region_id
city				city
city_id			City_id	city_id
population				population

Quindi, per arrivare al nostro obiettivo dobbiamo creare manualmente una nuova classe globale chiamata “CUSTOMER”, dove aggiungeremo tutti gli attributi di nostro interesse:

Mapping Table: CUSTOMER

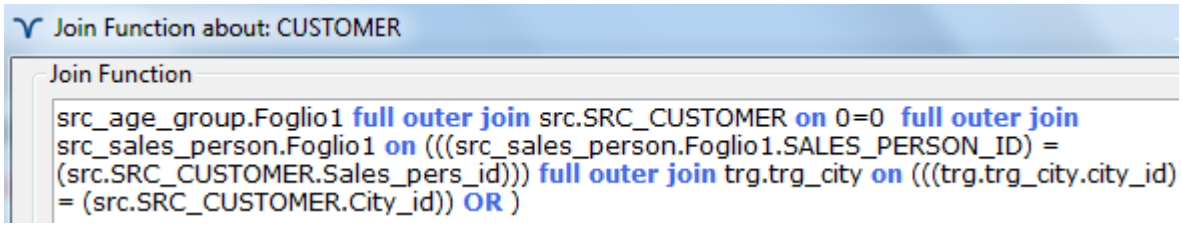
CUSTOMER(globalSource)	Foglio1(src_age_group)	Foglio1(src_sales_person)	SRC_CUSTOMER(src)	trg_city(trg)
Address			Address	
Age			Age	
City_id			City_id	city_id
Custid			Custid	
Dear			Dear	
Phone			Phone	
Sales_pers_id		SALES_PERSON_ID	Sales_pers_id	
city_name				city
customer_name			First_name , Last_name	
max	AGE_MAX			
min	AGE_MIN			
seller_name		FIRST_NAME , LAST_NAME		

Global Source

- globalSource
  - CUSTOMER
    - Address [string]
      - src.SRC_CUSTOMER.Address [string]
    - Age [double]
      - src.SRC_CUSTOMER.Age [double]
    - City_id [double]
      - src.SRC_CUSTOMER.City_id [double]
      - trg.trg_city.city_id [double]
    - Custid [double]
      - src.SRC_CUSTOMER.Custid [double]
    - Dear [string]
      - src.SRC_CUSTOMER.Dear [double]
    - Phone [string]
      - src.SRC_CUSTOMER.Phone [string]
    - Sales_pers_id [string]
      - src.SRC_CUSTOMER.Sales_pers_id [double]
      - src_sales_person.Foglio1.SALES_PERSON_ID [string]
    - city_name [string]
      - trg.trg_city.city [string]
    - customer_name [string]
      - src.SRC_CUSTOMER.First_name [string]
      - src.SRC_CUSTOMER.Last_name [string]
    - max [double]
      - src_age_group.Foglio1.AGE_MAX [string]
    - min [double]
      - src_age_group.Foglio1.AGE_MIN [string]
    - seller_name [string]
      - src_sales_person.Foglio1.FIRST_NAME [string]
      - src_sales_person.Foglio1.LAST_NAME [string]

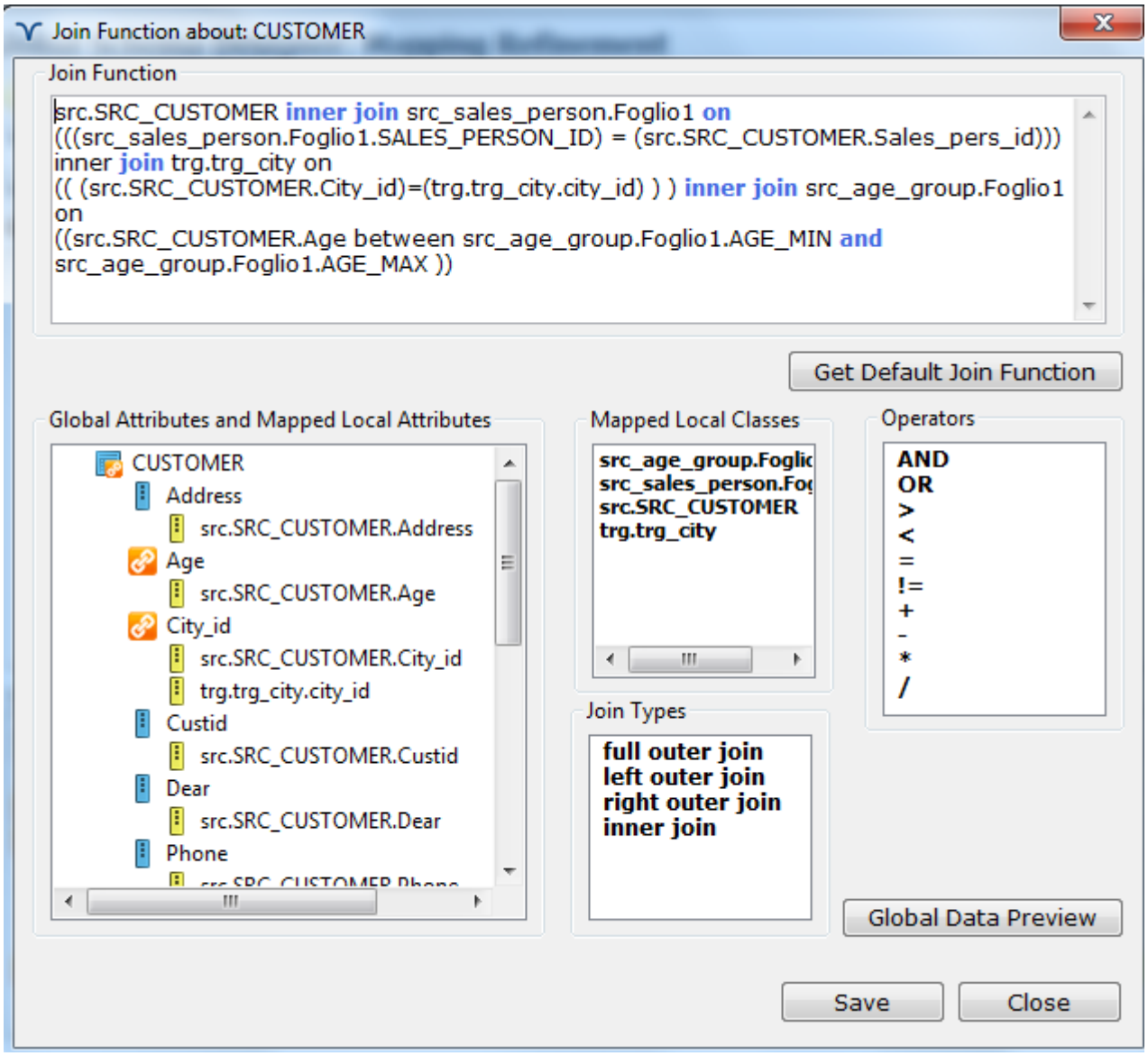
Abbiamo impostato come attributi di join “SALES_PERS_ID” , “CITY_ID” , “AGE_MIN”, “AGE_MAX” , “AGE” ed abbiamo anche modificato l’impostazione di join che veniva impostata automaticamente (non era corretta) ed aggiunto i vincoli per l’età.

Funzione di join creata in automatico da MOMIS (errata):

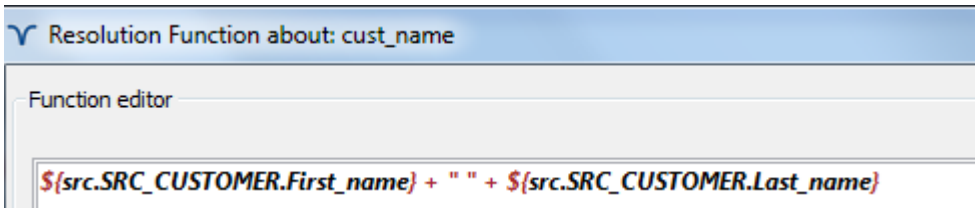


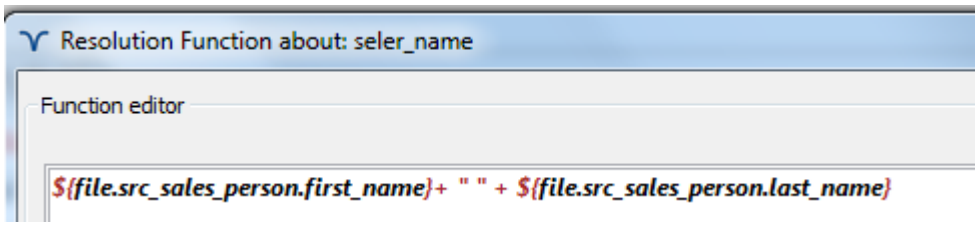
Funzione di join corretta ed impostata manualmente:

```
src.SRC_CUSTOMER inner join src_sales_person.Foglio1 on
(((src_sales_person.Foglio1.SALES_PERSON_ID) = (src.SRC_CUSTOMER.Sales_pers_id)))
inner join trg.trg_city on
(( (src.SRC_CUSTOMER.City_id)=(trg.trg_city.city_id) ) ) inner join  src_age_group.Foglio1 on
((src.SRC_CUSTOMER.Age between  src_age_group.Foglio1.AGE_MIN and
src_age_group.Foglio1.AGE_MAX ))
```

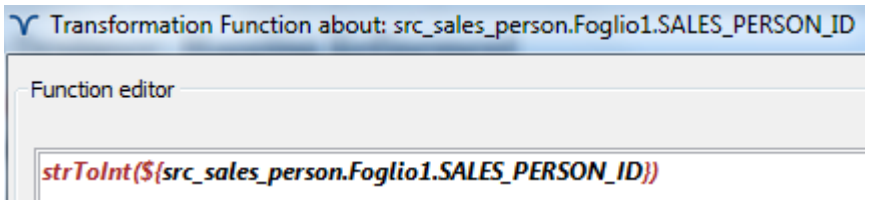


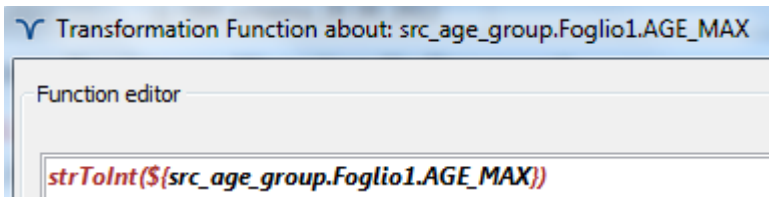
Abbiamo inoltre inserito le funzioni di Risoluzione, per unificare i nomi e cognomi dei clienti e dei venditori

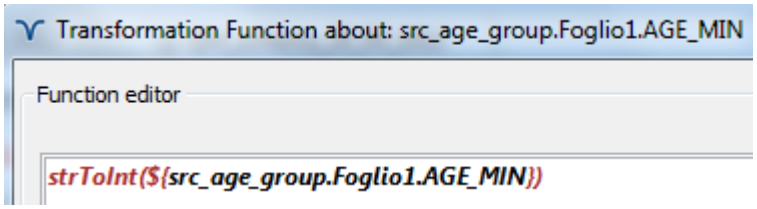




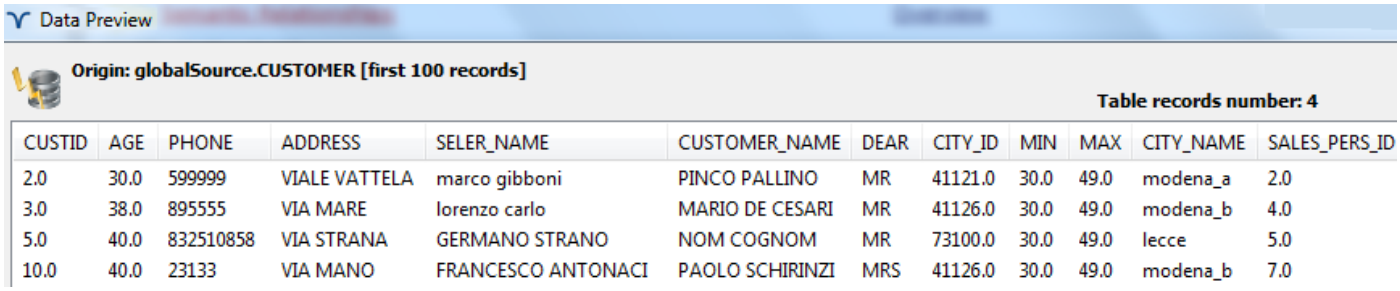
E funzioni di Trasformazione per effettuare il CAST dei vari campi provenienti dai file Excel (in quanto sono letti come STRING) in DOUBLE:





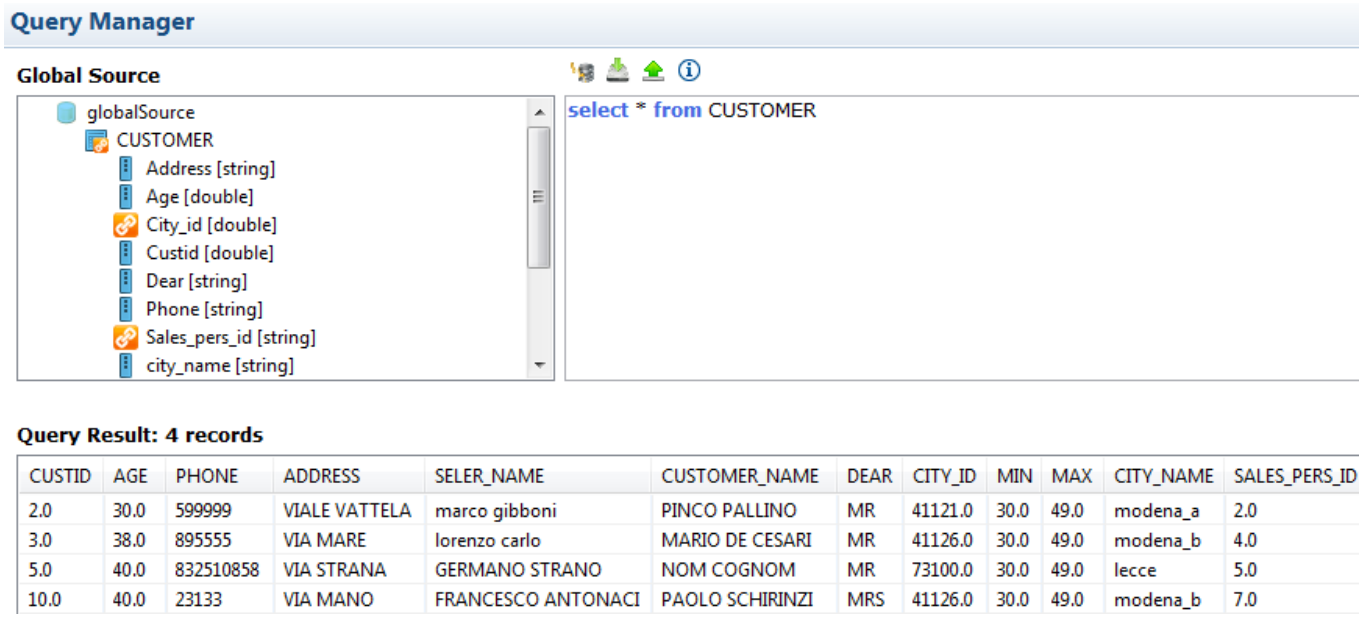


E questo è il risultato della "DATA PREVIEW":



CUSTID	AGE	PHONE	ADDRESS	SELER_NAME	CUSTOMER_NAME	DEAR	CITY_ID	MIN	MAX	CITY_NAME	SALES_PERS_ID
2.0	30.0	599999	VIALE VATTELA	marco gibboni	PINCO PALLINO	MR	41121.0	30.0	49.0	modena_a	2.0
3.0	38.0	895555	VIA MARE	lorenzo carlo	MARIO DE CESARI	MR	41126.0	30.0	49.0	modena_b	4.0
5.0	40.0	832510858	VIA STRANA	GERMANO STRANO	NOM COGNOM	MR	73100.0	30.0	49.0	lecce	5.0
10.0	40.0	23133	VIA MANO	FRANCESCO ANTONACI	PAOLO SCHIRINZI	MRS	41126.0	30.0	49.0	modena_b	7.0

Quindi il risultato della query "*Select * from CUSTOMER*" è:



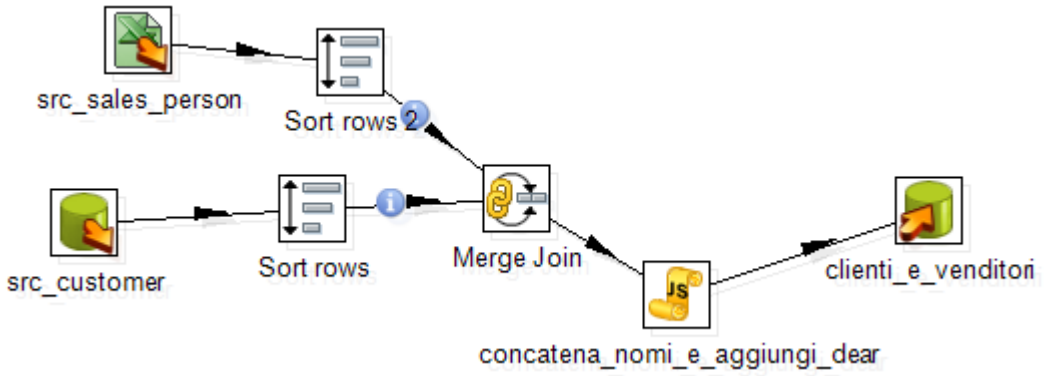
Query Result: 4 records

CUSTID	AGE	PHONE	ADDRESS	SELER_NAME	CUSTOMER_NAME	DEAR	CITY_ID	MIN	MAX	CITY_NAME	SALES_PERS_ID
2.0	30.0	599999	VIALE VATTELA	marco gibboni	PINCO PALLINO	MR	41121.0	30.0	49.0	modena_a	2.0
3.0	38.0	895555	VIA MARE	lorenzo carlo	MARIO DE CESARI	MR	41126.0	30.0	49.0	modena_b	4.0
5.0	40.0	832510858	VIA STRANA	GERMANO STRANO	NOM COGNOM	MR	73100.0	30.0	49.0	lecce	5.0
10.0	40.0	23133	VIA MANO	FRANCESCO ANTONACI	PAOLO SCHIRINZI	MRS	41126.0	30.0	49.0	modena_b	7.0

PENTAHO Data Integration

Abbiamo provato ad integrare i dati di questo esempio anche tramite Pentaho: per prima cosa possiamo affermare Pentaho Data Integration NON è compatibile con i File di Excel 2010.

Il primo passo della trasformazione è stato l'unificazione fra il DataBase che conteneva la tabella SRC_CUSTOMER (Clienti) ed il File parametro SRC_SALES_PERSON contenente i dati riguardanti i venditori.



Abbiamo fatto uso di uno script Java che porta in output delle variabili contenenti la concatenazione dei nomi e cognomi dei venditori e dei clienti, dopo che un "Merge Join" aveva effettuato un "INNER JOIN" fra le due sorgenti di dati sull'elemento "SALES_PERSON_ID". Abbiamo anche effettuato il controllo sul campo "DEAR" , in modo che possa impostare il valore di un'altra variabile in uscita (se dear=0=>dear_v="MR", =1=>"MRS",altro=>"MS").

Il risultato è stato inserito in una tabella ausiliaria.

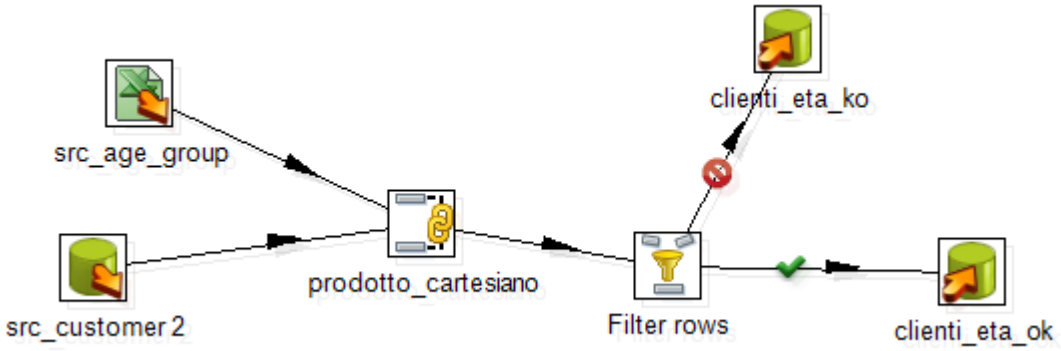
Nome del passo: concatena_nomi_e_aggiungi_dear

Java script:

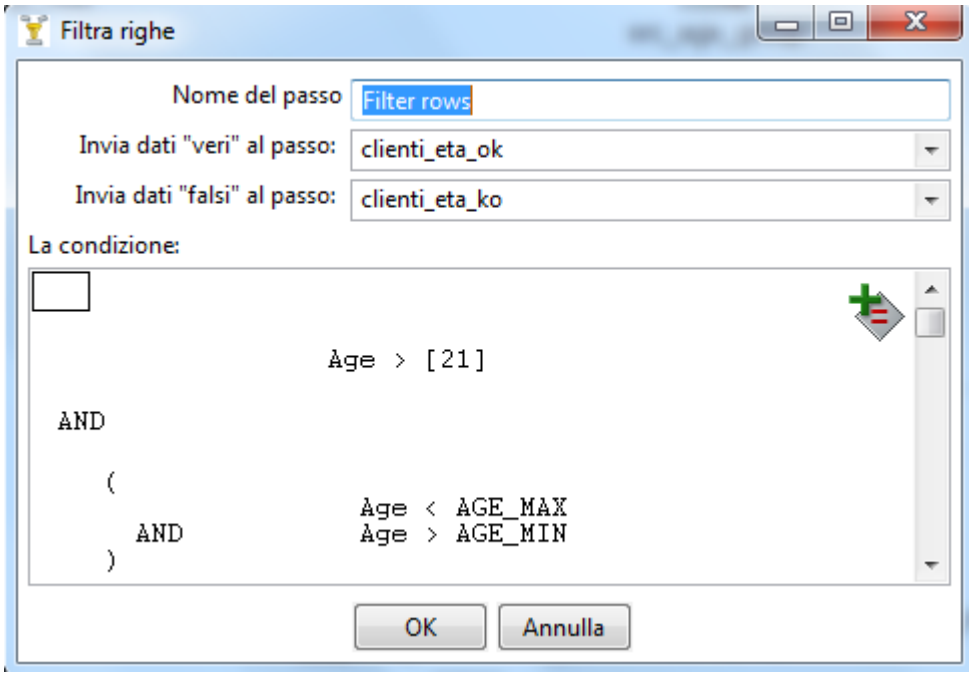
Script 1

```
//Script qua

var cust_name;
var sales_pers;
var dear_v;
var cre_date;
var upd_date;

cust_name=FIRST_NAME_1.getString()+ " "+ LAST_NAME_1.getString();
sales_pers=First_name.getString() + " " + Last_name.getString();

if( Dear.getInteger() == 0) dear_v="MR"; else
if( Dear.getInteger() == 1) dear_v="MRS"; else dear_v="MS";
```

Il secondo passo di aggregazione riguarda l'età dei clienti: deve essere compresa fra l'età massima e quella minima presente sul file paramento SRC_AGE_GROUP, ricordandoci poi che c'è anche la limitazione che i clienti non devono avere meno di 21 anni.

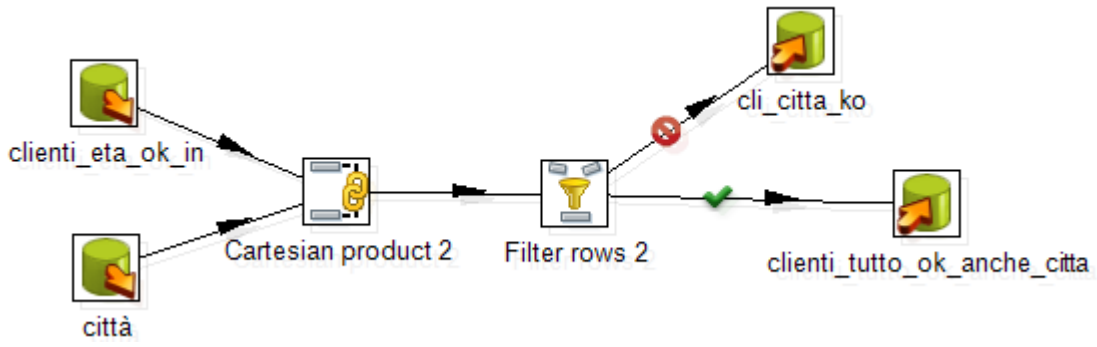


Il cui filtro è impostato nel seguente modo:



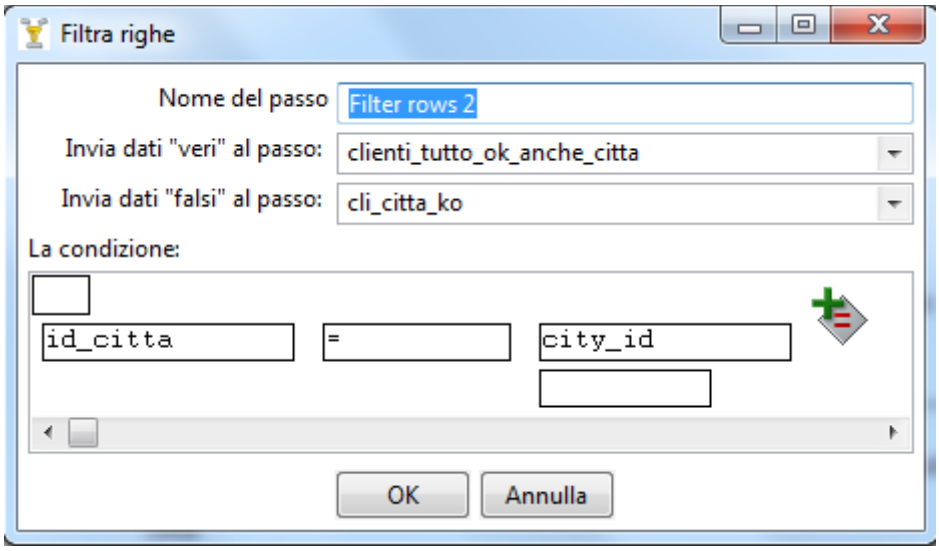
Le righe che superano questi requisiti sono salvate su un'altra tabella ausiliaria, in attesa di effettuare l'ultimo Step (e quindi per aggiungere gli ultimi vincoli).

L'ultimo vincolo è che la Città (CITY_ID) del cliente sia presente nella tabella TRG_CITY del DataBase TRGDEMO ed è stato ottenuto nel seguente modo:



Abbiamo effettuato un prodotto cartesiano fra tutti i “Clienti” fino ad ora selezionati (limite età) con il DB contenente tutte le Città , per poi filtrare solo le righe dove :

clienti_selez.city_id == città.city_id



Le righe così ottenute le abbiamo inserite in un’altra tabella momentanea, per poi selezionare i campi di nostro interesse e dirigerli verso la destinazione => TRG_CUSTOMER.

```
select * from CUSTOMER
```

Results | Messages

Custid	Address	City_id	Phone	Age	city	dear_v	cust_name	sales_pers	SALES_PERSON_ID
3	VIA MARE	41126	895555	38	modena_b	MR	lorenzo carlo	MARIO DE CESARI	4
10	VIA MANO	41126	23133	40	modena_b	MRS	FRANCESCO ANTONACI	PAOLO SCHIRINZI	7
5	VIA STRANA	73100	832510858	40	lecce	MR	GERMANO STRANO	NOM COGNOM	5

Si può notare una riga del risultato in meno rispetto a quanto ottenuto in MOMIS, ciò è dovuto al limite diverso impostato sull’età: BETWEEN impostato su MOMIS mi prende anche il risultato uguale al limite (30 risulta compreso fra 30 e 49) invece con Pentaho abbiamo impostato solo la relazione < e > ma non uguale

Con questo esempio abbiamo ancora una volta avuto la conferma su quali sono i punti di forza di questo software: l’estrema compatibilità con le sorgenti di dati e la grande varietà di strumenti che ci è messa a disposizione; D’altra parte però abbiamo mostrato come con Pentaho Data Integration ci si impieghi molto più tempo per arrivare ad un’integrazione: ciò è dovuto sempre al fatto che non ci sono strumenti automatici capaci di semplificare il lavoro dell’utente.

PROBLEMI RILEVATI ED ALCUNI WARNINGS

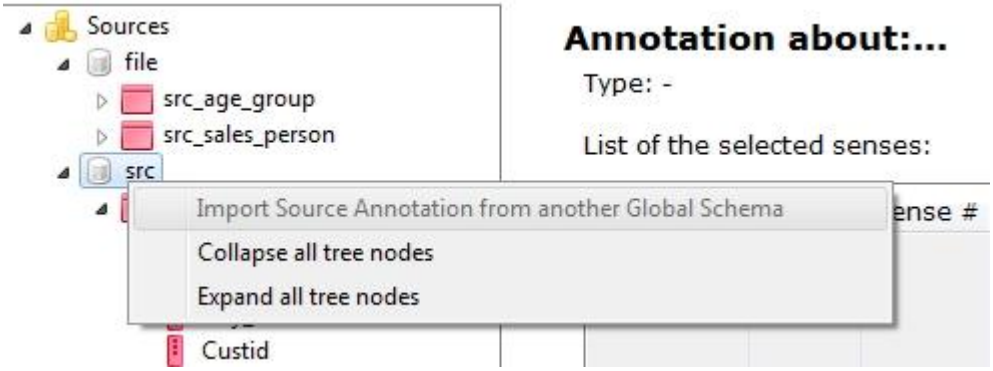
MOMIS:

PROBLEMI

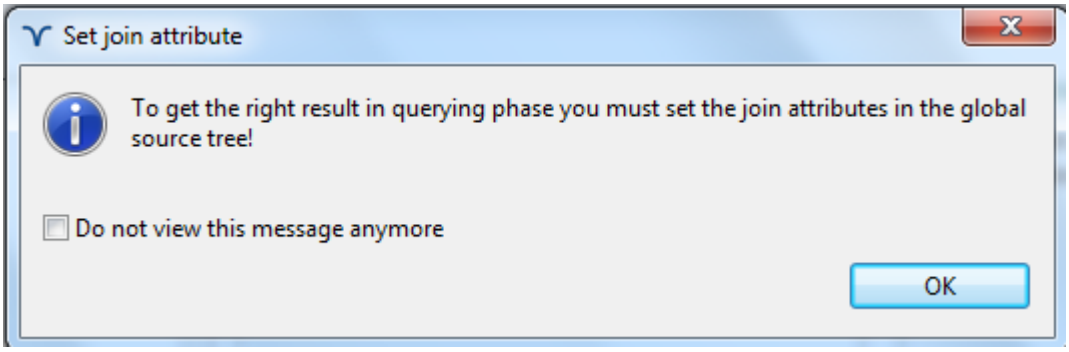
- Non c'è la possibilità di importare tabelle che contengano nel nome lettere accentate (ò,à,ù,ecc..) , il software le rileva ma non permette di aggiungerle nel primo passo dell'integrazione; Questo è un grosso problema.
- Capita che vengano create annotazioni automatiche di tutti gli elementi presenti con "NULL" => "a quantity of no importance, ecc..".
- Si blocca durante l'utilizzo di WordNetExtender => nella procedura di "Insert a new Synset" , "Search hypernim Synset" e per visualizzare l' "Hypernym Chart".

WARNINGS

- Non mi permette di importare le annotazioni. (Risolto: devo aprire insieme, nello stesso progetto, i vari Schemi Globali fra cui voglio effettuare l'importazione)



- L'integrazione di dati non è reale: è solo una "Vista", non ho la possibilità di "materializzarla" se non tramite l'ausilio di altri programmi.
- L'esecuzione delle query è molto lenta se la mole di dati da fornire in output è elevata (ordine di 10^3 risultati) in quanto l'aggregazione non è "fisica" ma è una "VISTA" (i dati continueranno a trovarsi nelle sorgenti); In alcune occasioni, se la query è "molto complicata" (es. query su un' integrazione di viste e DB) il risultato potrebbe essere calcolato anche in più di 2 minuti.
- Bisogna ricordarsi di settare come "attributi di join" nell'albero del global schema i campi che fanno parte della join function. Momis mi ricorda di fare ciò tramite un messaggio (quando si salva la join function modificata). Questo avviene perché attualmente non c'è un parser per la join function.



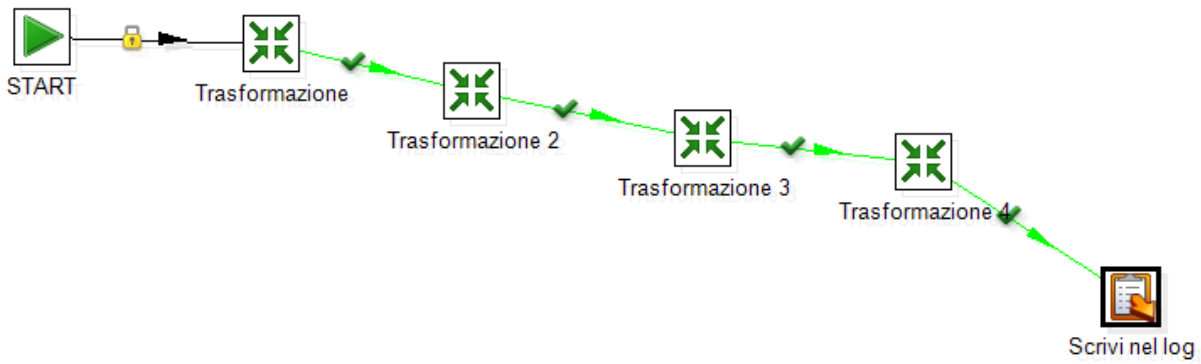
Pentaho Data Integration:

PROBLEMI

- La funzione “Anteprima” dei dati in input spesso non funziona;
- L’utilizzo del menù delle variabili d’ambiente con la versione 4.3.0 “beta” non era possibile su vari computer. L’unico metodo per richiamare la schermata era la combinazione di tasti “CTRL+SHIFT” => ma questa combinazione è assegnata anche al cambio lingua.
  **Problema risolto parzialmente con la versione definitiva di PDI 4.3.0, che ha aggiunto la possibilità di chiamare questa impostazione da “Modifica=>Imposta Varibili D’ambiente”, ma comunque resta impossibile inserirle per l’utilizzo nelle funzioni (es. su pc Asus questa combinazione di tasti è riservata per cambiare la lingua della tastiera).

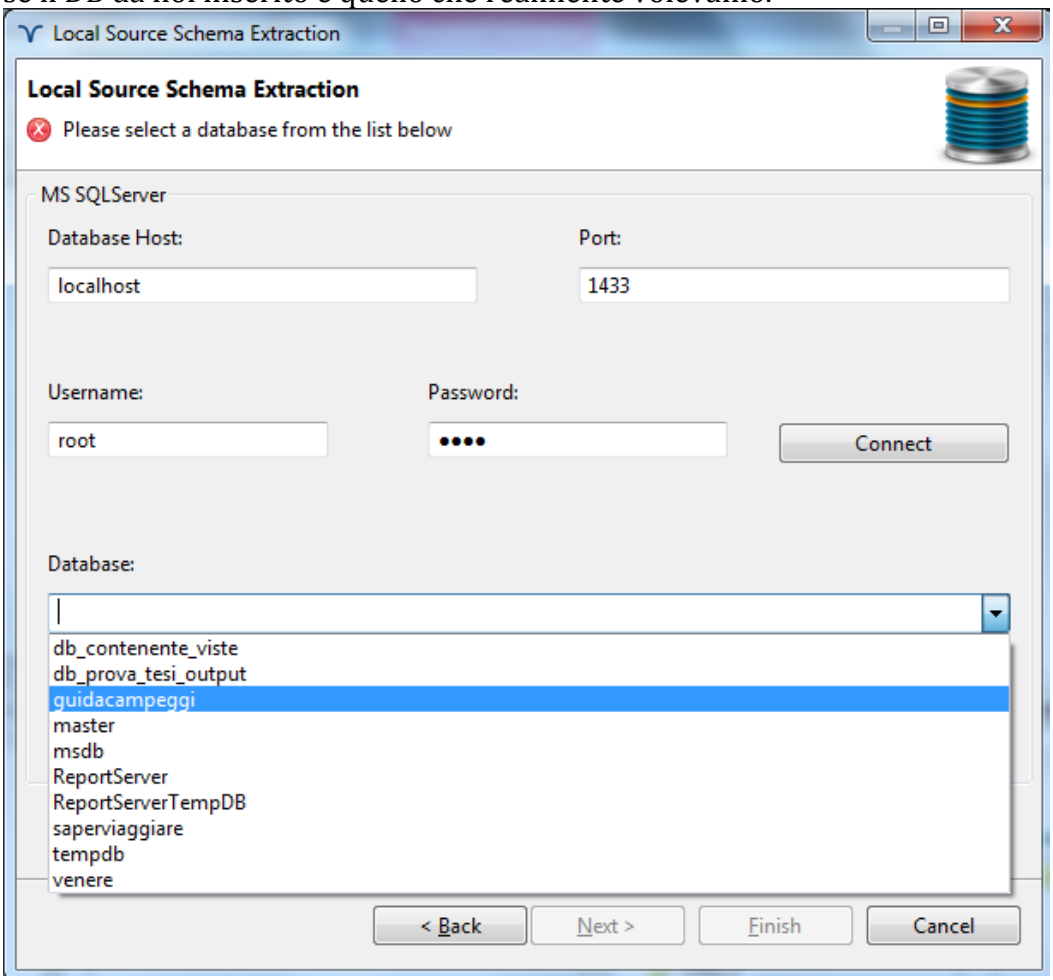
WARNINGS

- Se si modificano le strutture dati delle tabelle durante l’utilizzo del tool di programmazione grafica (Spoon) si possono avere degli errori. Essi sono causati dal fatto che Spoon utilizza una cache per non rileggere ogni volta dal database... basta ripulire la cache dal menu principale: “Strumenti->Database->Pulisci Cache”.
- (Lavorando con le Trasformazioni) Per ottenere un Aggregazione di Dati può capitare di collegare più passi in sequenza. Bisogna ricordare che essi vengono eseguiti in parallelo, quindi possono fornirci risultati errati: per esempio, abbiamo due elementi di base collegati fra loro (Ta=>Tb) , che fanno utilizzo di una stessa connessione al DataBase ; Quando A avrà “terminato” la sua funzione , verrà eseguito B , ma in realtà quando inizia B, A starà ancora lavorando sul DB , e B arriverà ad un risultato sbagliato (oppure genererà un errore nel caso che A gli avesse chiuso la Connessione). Un modo semplice per aggirare questo problema è la creazione di un Job composto da tante Trasformazioni “semplici” (che eseguono un solo passo) collegate fra loro.



- Problema (Risolto Grazie al forum di supporto): la connessione con i Database presenti in MySQL risulta impossibile causa errore “Exception while loading class org.gjt.mm.mysql.Driver” causato dalla mancanza del file “mysq-connector-java-5.1.17.jar” nella directory principale di Pentaho Data Integration.

- Si può notare che, quando andiamo a creare il collegamento con la Risorsa, dobbiamo ricordarci il nome preciso: Pentaho non ha un menù a tendina per la selezione (Momis invece offre questa comodità) ed oltretutto il pulsante "Esplora" serve solo per vedere se il DB da noi inserito è quello che realmente volevamo.



CONCLUSIONI

Dopo aver testato i due programmi possiamo affermare che **MOMIS** è indirizzato a degli utenti che vogliono arrivare ad un risultato in pochi passaggi , in quanto l'integrazione di dati avviene in maniera semi-automatica. Si deve intervenire manualmente solo per aggiustare e/o migliorare e raffinare il risultato dell'integrazione ; Invece, per chi ha la necessità di dover creare un integrazione di dati partendo da sorgenti molto varie, come per esempio da file di varia natura, eMail, ecc.. senza voler utilizzare altri software per fare un primo passo di "omogeneizzazione" (in sorgenti di dati che possono essere integrate con MOMIS) , la scelta ricade su **Pentaho Data Integration** , che ci fornisce moltissimi strumenti per farlo . Inoltre, se abbiamo la necessità di effettuare studi di Business Intelligence (**BI**) , quindi raccogliere e analizzare questi dati, è consigliabile lavorare direttamente con Pentaho D.I. , in quanto esso mette a disposizione vari Tools sviluppati specificamente per questo lavoro (Pentaho Business Analytics) .

	MOMIS	PENTAHO D. I.
Facilità d'utilizzo	✓ ✓ ✓	✓
Funzionalità	✓	✓ ✓ ✓
Conoscenze necessarie per l'utilizzo	✓ ✓	✓ ✓ ✓
Forum di Supporto		✓ ✓
Tutorial	✓ ✓ ✓	✓
Materiale on-line	✓	✓ ✓ ✓

Funzioni disponibili	MOMIS	PENTAHO D.I.
Importazione Risorse	**SI** (DB2, Microsoft SQLServer , MySQL , Oracle , PostgreSQL Database, Sorgenti JDBC / JDBC-ODBC , File Excel, File CSV, Altre risorse tramite WEB)	**SI** (oltre 25 DB oltre che varie tipologie di file)
Utilizzo Risorse Remote	**SI**	**SI**
Plug-in per introdurre nuove funzionalità	**NO**	**SI**
Progetto dimostrativo	**SI**	**NO**
Integrazione basata su relazioni semantiche	**SI** (WordNet)	**NO** (Weka [4])
Possibilità di interrogare il risultato dell'integrazione	**SI** (strumento integrato)	**NO** (Sever DBMS esterno)
Materializzazione risultato	**NO**	**SI**

[4] Weka => Data Mining Software in Java (http://weka.pentaho.com/)

Bibliografia

[1] S. Bergamaschi, D. Beneventano, A. Corni, E. Kazazi, L. Po, M. Orsini, S. Sorrentino (2011,Giugno). The Open Source release of the MOMIS Data Integration System. Proceedings of the Nineteenth Italian Symposium on Advanced Database Systems (SEBD 2011), Maratea, Italy, pp.175-186 .

[2] C. Fellbaum (2005). WordNet and Wordnets. In E. O. Linguistics. Oxford: Elsevier.

[3] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller (1990). WordNet: An on-line lexical database. International Journal of Lexicography.

[4] *Oracle*® (2008, Settembre). *Data Integrator:Getting Started with an ETL Project.*

[5] M. Ricci (2009) LA BUSINESS INTELLIGENCE - GESTIONE DI UN EVENTO COMMERCIALE : IL CASO CREDITO EMILIANO , Tesi di Laurea.