*Università degli Studi di Modena e Reggio Emilia*

---

Facoltà di Ingegneria – Sede di Modena

Corso di Laurea in Ingegneria Informatica – *Nuovo Ordinamento*

# Analisi e sperimentazione del componente software IBM – DB2 Information Integrator

Relatore:
Prof. Sonia Bergamaschi

Candidato:
Alberto Fontanesi

Correlatore:
Prof. Maurizio Vincini

Anno Accademico 2002-2003

# Indice

# Introduzione

## 1.1 L'integrazione dell'informazione

Negli ultimi anni si sta assistendo ad un fenomeno che si sta facendo sempre più rilevante, si tratta del tentativo delle imprese di avvicinarsi al mondo on demand, ovvero lo sforzo di possedere la capacità di attuare processi economici, siano essi interni o con terzi, che rispondano con rapidità ad ogni richiesta del consumatore, ad ogni opportunità offerta dal mercato e alle eventuali minacce esterne.
Per raggiungere tale traguardo un'impresa necessita di una infrastruttura dedicata alla tecnologia dell'informazione (meglio conosciuta come information technology, IT) le cui priorità devono essere allineate con gli obiettivi aziendali e che ha il compito di favorire il miglioramento delle operazioni commerciali rendendole più flessibili e maggiormente ricettive; questa infrastruttura, che prende il nome di ambiente operativo on demand (on demand operating enviroment), è progettata in modo da aiutare le imprese a ridurre i costi, migliorare l'assetto delle risorse utilizzate e indirizzarle verso nuove opportunità commerciali.
Come mostrato in figura 1 questo ambiente, che altro non è che una piattaforma integrata, è composto da tre caratteristiche principali:

| | |
|---|---|
| Integrazione | migliora l'efficienza e la combinazione flessibile di risorse in modo da ottimizzare le operazioni sia interne che esterne all'azienda. |
| Automatizzazione | riduce la complessità del sistema manageriale permettendo un miglior utilizzo delle risorse, ottimizzandone la disponibilità e l'elasticità, e abbattendo i costi relativi a obiettivi o politiche aziendali. |
| Modellizzazione | fornisce una visione unica e consolidata delle risorse disponibili senza preoccuparsi di dove esse risiedano fisicamente. |

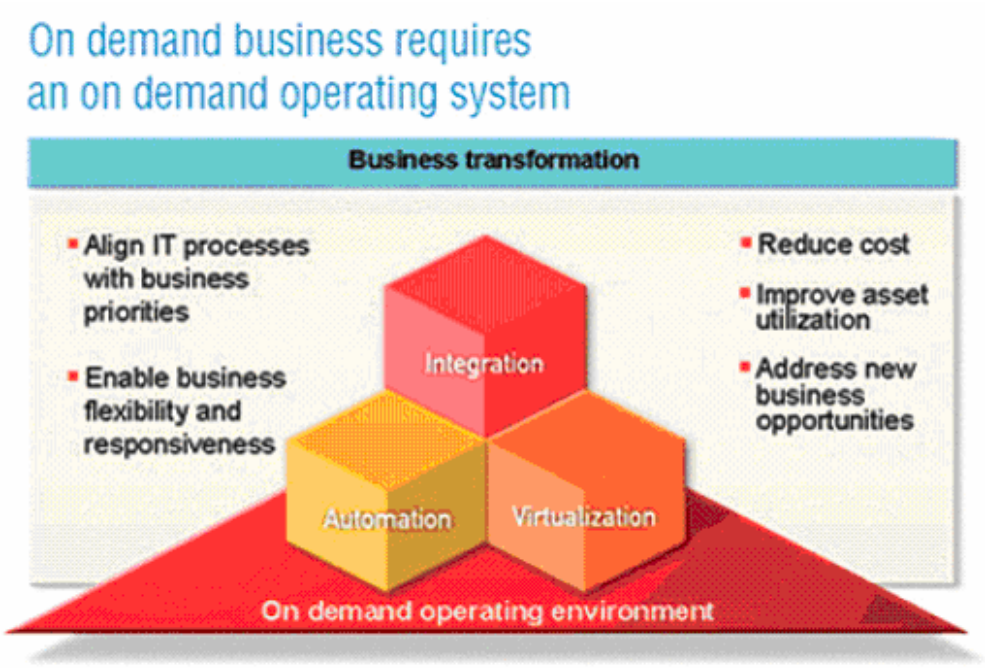



**Figura 1 – L'ambiente operativo on demand**

La realizzazione di un'informazione integrata sia all'interno che all'esterno dell'azienda è un compito arduo, operazioni come la gestione delle relazioni coi clienti, della catena di approvvigionamenti, della distribuzione sono basate sulla realizzazione dell'integrazione di informazioni provenienti da sorgenti diverse le quali possono essere o meno strutturate e che difficilmente si prestano ad essere copiate o riunite in un unico database.

Da qui si deduce un altro grosso scoglio che si incontra nel conseguimento di questo obiettivo ovvero l'eterogeneità delle sorgenti e dei dati.
La digitalizzazione delle informazioni è in rapida crescita e apparentemente supera la capacità del mercato di gestirla e di influenzarla. Gli analisti del settore industriale della School of Information Management and System dell'università della California avevano stimato che i dati che si sarebbero generati tra il 2001 e il 2003 sarebbero risultati di gran lunga più numerosi di tutti quelli prodotti nel corso della storia dell'informatica. Questa previsione è stata ampliamente confermata; si pensi che attualmente vengono elaborati tra gli 1 e i 2 exabyte (1 e 2 miliardi di gigabyte) di informazioni all'anno il che significa che approssimativamente ogni individuo produce in media 250 megabyte.[1]
Risulta inoltre chiaro che nell'era di internet, dell'ampliamento dei mercati che si fanno via via sempre più estesi e complessi non si ci può aspettare il ricorso a semplici e tradizionali applicazioni come può essere, ad esempio, l'uso di un database relazionale ma è necessario considerare anche altre tipi di fonti quali ad esempio i documenti XML (Extensible Markup Language), documenti di testo, video clip, immagini, file Excel, Web content, e-mail, banche dati di notizie, special-pourpose stores sia interni che esterni che rendono, come già detto, molto più gravoso il reperimento delle informazioni effettivamente necessarie.
Queste massicce quantità di dati già disponibili ed accessibili dalle imprese, si presentano in una forma frammentata e disaggregata che si affianca ad una forte e sempre più frequente richiesta, da parte delle aziende, di una visione unificata dell'informazione; è proprio in questo contesto che trova origine DB2 Information Integration il cui principio base è, appunto, quello di permettere agli utenti di prescindere dalle molteplici sorgenti su cui possono risiedere i dati e di poterli trattare come se fossero tutti presenti su di una unica, nascondendo tutte le complessità derivanti dall'utilizzo d'informazioni memorizzate in differenti locazioni, linguaggi, formati e con metodi d'accesso eterogenei. Utilizzando linguaggi di base come SQL o XML (attraverso XQuery), un Web service standard o un content API, l'integrazione dell'informazione facilita agli utenti l'accesso trasparente ai dati senza doversi preoccupare né della loro effettiva implementazione fisica, né di aspetti riguardanti consistenza, integrità e sicurezza.

### *1.1.1 Caratteristiche principali*

Fra le tante caratteristiche possedute dall'integrazione dell'informazione ve ne sono alcune sicuramente di grande importanza. L'IBM, come si vedrà successivamente, è una delle società che maggiormente ha investito in questo campo ed essa stessa individua e delinea i tratti fondamentali di questa infrastruttura così come qui di seguito descritto e come rappresentato in figura 2.

- **Federazione** L'IBM rappresenta l'azienda leader in questo settore. Il meccanismo di federazione permette di poter considerare e gestire diverse sorgenti di dati come se fossero una sola, garantendo, tra l'altro, il mantenimento della loro autonomia e integrità. Le proprietà di questa caratteristica sono:
- *Trasparenza:* aiuta a nascondere all'utente le molteplici differenze, peculiarità e implementazioni delle varie sorgenti dati creando astrazioni che permettono, virtualmente, di considerarle appartenenti ad un unico sistema.

[1] P. Lyman, H. Varian, J. Dunn, A. Strygin, K. Swearingen, "How Much Information?", University of California, Berkeley, Ottobre 2000, http://sims.berkeley.edu/research/projects/how-much-info/

- *Eterogeneità:* caratterizzata dalla capacità di federare tipi di dati altamente diversificati tra loro (dati strutturati, semi strutturati e non strutturati).
- *Estensibilità:* intesa come la possibilità di estendere il processo di federazione quasi ad ogni tipo di sorgente. Nello specifico, questa caratteristica, è stata progettata per minimizzare gli sforzi necessari per l'integrazione di una nuova sorgente; per il momento offre la capacità di fornire le informazioni necessarie per ottimizzare l'accesso da parte delle query.
- *Piena funzionalità:* comprende le funzioni disponibili tra i linguaggi di query supportati compensando quelle mancanti, in più fornisce la possibilità di includere capacità specifiche della sorgente all'interno del linguaggio d'interrogazione utilizzato.
- *Autonomia delle sorgenti:* che permette loro di poter essere federate sia con un modesto che con considerevole impatto sui sistemi o sulle applicazioni esistenti.

- **Ricerca** L'infrastruttura dell'IBM mette a disposizione degli utenti avanzate capacità di ricerca e d'interrogazione includendo la possibilità di esplorare il Web, indici di documenti, risultati di ricerche federate ottenuti da svariati motori di ricerca, catalogare e riassumere documenti di testo per consentirne l'accesso intelligente e comprenderne i significati. Nel 2002 la società ha dato vita all'IBM Search and Text Institute allo scopo di riunire ed accelerare i propri studi e il proprio dispiegamento di funzioni di ricerca avanzata e di estrapolazione in un'architettura complessa; i risultati ottenuti alimentano la piattaforma di DB2 Information Integrator così come le altre offerte della compagnia.

- **Immagazzinamento temporaneo** Per permettere appropriate caratteristiche di prestazione, diffusione e disponibilità a favore delle applicazioni richiedenti è necessario il ricorso ad una serie di strategie di memorizzazione. L'IBM supporterà il posizionamento e la gestione delle informazioni in più punti della gerarchia di dati al fine di incrementare le performance del sistema. Quella appena descritta non è una semplice caching, ma è una politica molto più complessa basata sul data management e sul data placement.

- **Trasformazione** Sono previste caratteristiche di trasformazione molto evolute per facilitare le operazioni di analisi, presentazione ed interscambio.

- **Riproduzione** Rappresenta una caratteristica fondamentale di ogni infrastruttura d'integrazione dell'informazione in quanto completa le funzionalità dell'accesso distribuito, favorisce la gestione di data stores centralizzati, e fornisce gli strumenti necessari per gestire cache di dati efficientemente.


**Figura 2 – Caratteristiche principali dell'integrazione dell'informazione**

## 1.2 Metodi d'accesso ai dati

Finora è stato esaminato, a livello macroscopico, l'attuale situazione della dislocazione delle informazioni nonché le problematiche che essa comporta. Ora si passerà ad esaminare i metodi di reperimento dei dati analizzandone contemporaneamente sia gli aspetti principali sia le difficoltà che essi incontrano nell'adempire alla loro funzione.
Da qui in poi la trattazione si focalizzerà in maniera ancora più assidua sul pensiero dell'IBM, la quale punta a ridurre se non addirittura ad eliminare le problematiche esposte in precedenza in modo da comprendere ancora più efficacemente le motivazioni che hanno spinto tale società a realizzare DB2 Information Integrator.

Attualmente, per il reperimento delle informazioni, esistono due approcci: uno largamente più diffuso, l'altro da poco affacciatosi sul mercato.
Il primo si compone di quattro operazioni extract, transform, load (ETL) e replication e prende il nome di data placement o consolidation; il secondo, invece, è conosciuto come distribuited access, enterprise information integration (EII) o federation e permette l'accesso diretto ai dati nelle loro locazioni originarie; questi metodi possono operare sia distintamente che parallelamente a seconda delle necessità.
In una visione molto semplificata federation scompone le query in sotto-query appropriate ognuna rivolta ad una differente sorgente in base alla locazione dei dati necessari ottenendo così il risultato desiderato; il data placement, invece, trasferisce le informazioni presenti nelle varie sorgenti raggruppandole ed eseguendo, solo dopo averle memorizzate localmente, le operazioni richieste senza disgregare le query

Entrambi gli approcci necessitano di strumenti quali *mapping* e *trasformation* che garantiscono il mantenimento dell'integrità e della consistenza dei dati.
Il mapping permette di comprendere le relazioni esistenti fra i diversi frammenti di dati e di possedere la capacità di determinare, ad esempio, che una colonna presente in una tabella virtuale in realtà corrisponde ad un attributo di un'altra sorgente d'informazioni.

Trasformation, invece, converte e combina i dati collegati tramite il mapping fra diverse rappresentazioni; come esempio si può pensare ad una semplice trasformazione da intero a stringa oppure ad una conversione di dati relazionali in dati gerarchici.
Esiste inoltre il *caching* che rappresenta un altro legame fondamentale tra i due modelli d'integrazione sopra esposti e che opera sfruttando i punti di forza di uno in modo da colmare le lacune dell'altro. Questo metodo realizza un temporaneo immagazzinamento dei dati che può incrementare le performance di federation attraverso la memorizzazione trasparente di una copia locale di un insieme di risultati; dal punto di vista del data placement esso non è altro che una possibile copia trasformata di qualche sorgente remota che potrebbe aver bisogno di essere manipolata.

È comunque necessario sottolineare che alla base della corretta funzionalità di questi strumenti deve essere presente una dettagliata e minuziosa descrizione dell'ambiente in cui operano la quale deve includere tra l'altro relazioni, locazioni, formati tecnici, in poche parole metadati.

### 1.2.1 Enterprise Information Integration: EII

Il nucleo di questo approccio si sviluppa attorno alle query cosiddette federate.
Il caso più semplice si riduce a considerare un'applicazione che richiede il collegamento di dati presenti in due diversi database, che possono essere, ad esempio, DB2 e Oracle. Dal punto di vista applicativo questa sarebbe una banale query contenente un join fra due tabelle che risulta essere, in realtà, molto più complessa a causa del fatto che una di queste tabelle è presente localmente su DB2 mentre l'altra in una locazione remota su Oracle. Il sistema deve, ovviamente, creare due query distinte adatte ai rispettivi DBMS (database management system) e decidere il criterio più efficiente per realizzare il join tenendo presente alcuni importanti fattori come le dimensioni relative dei set di risultati di ogni singola query, la capacità di elaborazione dei due sistemi, la velocità di connessione presente tra loro e così via. Comprensibilmente per richieste più complesse saranno necessarie maggiori capacità rispetto a quelle appena esposte; a molti dati, infatti, si può accedere tramite una vasta varietà di linguaggi o di applicazioni che raramente hanno in dotazione l'ampio set di funzioni disponibili in SQL. IL compito di federation sarà dunque quello di simulare le funzionalità richieste, collaborare nel processo di ottimizzazione delle query, eseguirle presso le opportune sorgenti e di svolgere un ruolo di mediatore fra i differenti metodi d'accesso prendendo in considerazione anche aspetti come la gestione della sicurezza e l'integrità transazionale. Risulta chiaro che tutte queste operazioni comportano una maggiore lentezza di esecuzione rispetto al data placement, ma è importante sottolineare che, se da un lato si incontra questo svantaggio, dall'altro EII presenta molti punti di forza, infatti è in grado di:

- Ridurre i costi d'implementazione e di manutenzione in quanto non necessita di hardware aggiuntivi, come memorie o server
- Accedere in tempo reale a dati risiedenti in sorgenti remote
- Elaborare congiuntamente informazioni in formati tradizionali e non
- Accedere a copie di dati protette da algoritmi di sicurezza, da restrizioni sull'accesso o da regolamenti societari che non consentono il trasferimento dei dati stessi (alcuni paesi europei, per esempio, non permettono il mescolamento di informazioni personali relative alla clientela con informazioni relative ai vari account in un singolo database; è però possibile, grazie a questo approccio, realizzare una singola immagine di questi dati federandoli al momento dell'accesso)

È importante sottolineare il fatto che, comunque, tutte queste diverse sorgenti devono essere sufficientemente consistenti al fine di ottenere aggregazioni effettivamente realizzabili e significative, deve esistere, infatti, una chiave attraverso la quale sia possibile raggruppare o correlare le informazioni (come, ad esempio, un custumer id).

Finora si è discusso unicamente sulle modalità di lettura dei dati, è necessario, però, tener presente che le potenzialità di questo approccio non sono ristrette solo a questo ambito ma si allargano anche alla possibilità di un accesso distribuito in scrittura.

### 1.2.2 ETL/replication

Il data consolidation rappresenta l'approccio tradizionale di integrazione dell'informazione; in esso si assiste, contrariamente al EII, ad un trasferimento dei dati da sorgenti disparate, in cui essi si trovano originariamente, a un'unica area locale sulla quale verranno interrogati. Considerato da sempre meno complesso del metodo federation questo sistema, una volta creata la copia locale delle informazioni necessarie, si sottrae al gravoso compito rappresentato dalla considerevole manipolazione dei dati nonché dalla gestione degli accessi alle locazioni remote presenti all'interno delle query generate dai singoli utenti. Molta importanza viene assunta anche dal tempo massimo di latenza che può essere tollerato durante un trasferimento da una sorgente ad una destinazione che comunque è solitamente noto all'interno di un determinato ambiente. Come esempio si può considerare un data warehouse nel quale la frequenza delle transazioni richiesta potrebbe essere giornaliera o settimanale mentre la latenza può facilmente estendersi ad alcune ore; al contrario, nel caso in cui siano necessari dati in tempo reale la latenza deve essere il più esigua possibile. Tale latenza minima è determinata da due importanti fattori rappresentati dalla complessità delle trasformazioni da eseguire e dal volume dei dati da trasferire; essi sono elementi caratterizzanti dei due approcci complementari che costituiscono il data placement. Mentre ETL è ottimizzato per elevati volumi di dati e spesso è anche utilizzato in operazioni che prevedono trasformazioni molto complesse, replication è indicato per il trasferimento di singoli record e per trasformazioni più semplici.

### 1.2.3 Federation VS Consolidation

Federation e data consolidation in realtà potrebbero considerarsi sinonimi, entrambi, infatti, prevedono richieste e ricezioni di dati che originariamente risiedono al di fuori dei confini fisici del database e coi quali gli utenti interagiscono. La differenza fondamentale è il momento in cui avvengono le richieste dei dati e del loro trasferimento nel database. Nell'approccio federation si verificano entrambe dopo che l'utente ha effettuato la sua richiesta; nel data consolidation, invece, avvengono in anticipo e, in pratica, la richiesta si effettua una volta durante la definizione del trasferimento mentre il trasferimento stesso può avvenire più volte. Dal punto di vista degli utenti o delle applicazioni, i due approcci si comportano in maniera completamente differente: il primo integra le informazioni richieste on the fly direttamente dalle loro ubicazioni originarie agendo su esse solo dopo che l'utente decide quali dati richiedere, il tutto in un lasso di tempo accettabile; il secondo opera anticipatamente rispetto alle richieste inviate assicurandosi, così, tempistiche maggiori nell'eseguire il processo a cui è adibito. Consolidation necessita, comunque, di conoscere in anticipo quali dati saranno trattati ed esige maggiori capacità in termini di memoria necessari a causa della copia locale dei dati che realizza. Bisogna fare attenzione a non incorrere nel facile errore di pensare che EII possa essere impiegato in modo diretto e non pianificato per fornire agli utenti un accesso a qualsiasi tipo di dato e in qualunque locazione. Se è vero che le query permettono di collegare e importare informazioni on the fly è necessario tener presente che ogni accesso incontrollato può comportare problemi significativi sia per gli utenti che per i sistemi IT.
Federation, di fatto, esige più rigorose analisi, modellazioni, controlli e pianificazioni rispetto al data placement, questo per sottrarsi da eventuali problemi semantici o riguardanti le prestazioni in cui gli utenti potrebbero imbattersi se provassero a trattare dati inconsistenti nel significato o nella struttura. I dati "consolidati", memorizzati come in un data warehouse, e i dati operazionali, costruiti attraverso il data placement, evitano questi problemi facendo sì che questo approccio continui a giocare un ruolo importante in questo settore.

### 1.2.4 Utilizzo concorrente di federation e data consolidation

Esistono situazioni in cui è necessario l'impiego di entrambi gli approcci, in alcuni casi, ad esempio, permettere ad un'applicazione l'accesso diretto a gruppi di dati potrebbe essere politicamente o tecnicamente irrealizzabile, in altri potrebbe semplicemente risultare oneroso il trattamento di dati memorizzati in strutture che rallenterebbero troppo le performance dell'intero

sistema. In questi casi federation può usare il data consolidation per creare o gestire cached data per agevolare il proprio operato. Attualmente queste informazioni trasferite localmente possono essere specificate esplicitamente e definite dai progettisti di sistema in anticipo, in futuro sarà, invece, la stessa tecnologia di integrazione dell'informazione che riuscirà automaticamente a configurarle nel modo appropriato.
Chiaramente questo approccio limita notevolmente le potenzialità del metodo EII, si pensi a cached data non totalmente aggiornati, o a come accessi in scrittura risulterebbero limitati nel caso in cui la cache non supportasse a pieno la sincronizzazione bilaterale, d'altra parte è necessario sottolineare che, comunque, esso amplia notevolmente l'insieme di soluzioni che federation riesce a fornire.
A questo punto ci si può chiedere come il data placement possa sfruttare i vantaggi offerti da dal distibuited access. Una delle principali caratteristiche di data consolidation è l'elevato numero e la smisurata varietà di sorgenti e di destinazioni attraverso le quali esso potrebbe scambiare i dati; generalmente, però, questi tool sono ottimizzati per interagire con un ristretto sottoinsieme di sorgenti e di destinazioni e l'utilizzo di federation può estendere questo subset mediante "pre-join" di dati dalle diverse sorgenti. Ovviamente questa intensificazione delle risorse in gioco incide sulle performance, sarà dunque necessario valutare bene le situazioni in cui è effettivamente vantaggioso il ricorso a questa metodologia tenendo comunque presente che la si può applicare solamente a casi in cui non si trattano elevati volumi di dati.

# DB2 Information Integrator

Il software IBM DB2 Information Integrator rappresenta la base per la realizzazione di un framework strategico d'integrazione dell'informazione che permette agli utenti l'accesso, la manipolazione e l'integrazione in tempo reale di dati tra loro diversi e distribuiti geograficamente.

Questo tool rientra facilmente nell'architettura service-oriented; ogni interrogazione SQL o stored procedure che interagisce coi dati remoti ha la capacità di restituire risultati in formato XML attraverso una richiesta ai Web service. Nella realtà questa trasformazione è resa possibile grazie all'IBM WebSphere Studio che automaticamente converte un'operazione SQL in un'operazione WSDL (Web service description language).

DB2 Information Integrator permette di ridurre i costi e i tempi di sviluppo di complessi processi d'integrazione WebSphere semplificando la progettazione, la realizzazione, e la manutenzione di collaborazioni che integrano sorgenti di dati diverse; in più agevola le attività manageriali tipiche di un'impresa grazie all'utilizzo di un sistema informativo che correla le informazioni sugli eventi coi relativi dati real-time provenienti dai sistemi di produzione, e/o con dati storici prelevati da data werehouse.

Se affiancato a WebSphere MQ, DB2 Information Integrator consente ai database administrator l'accesso a code di messaggi attraverso l'utilizzo del linguaggio SQL semplificando notevolmente l'integrazione fra database e sistemi di messaggistica.
Questo nuovo tool rappresenta una valida e semplice soluzione per la realizzazione e la pubblicazione di documenti XML generati attraverso dati reperiti da sorgenti differenti, infatti una singola query può accedere a tutte le informazioni necessarie indipendentemente da dove esse risiedano, convalidare il documento attraverso DTD (document type definition) o XML schema e pubblicarlo in una coda di messaggi.

DB2 Information Integrator può essere utilizzato con gli strumenti analitici o di resoconto delle aziende leeder da Business Objects, Brio, Cognos, Crystal Decisions, Microstrategy, SAS ed altri cooperando con i software esistenti senza ricorrere a procedure d'aggiornamento grazie all'accesso completamente trasparente ai dati così come confermato dagli studi dei partners ISV.
L'IBM ha portato a termine con successo i test di compatibilità di questo programma sia con WebSphere Studio che con Microsoft Visual Studio.Net garantendo, in questo modo, che i benefici riguardanti la produttività descritti in precedenza fossero applicabili sia alla tecnologia Java sia e quella .Net; i beta test hanno inoltre confermato la sua applicabilità all'ambiente PowerBuilder.

Tutto questo si concretizza in un ecosistema partner creato dalla società al fine di migliorare le capacità di analisi e di resoconto, esso permette un rapido sviluppo delle applicazioni, arricchisce ed accelera le funzioni di consegna ed estende l'accesso a sorgenti di dati addizionali

La famiglia di prodotti DB2 Information Integrator, che completa la famiglia di prodotti d'integrazione WebSphere, si compone, come mostrato in figura 3, di:

- IBM DB2 Information Integrator V8.1, un nuovo prodotto basato sulla tecnologia DB2 information management che rappresenta il middleware EII e replication
- IBM DB2 Information Integrator for Content V8.2, che rappresenta l'evoluzione dell'IBM Enterprise Information Portal

Ognuno di questi prodotti permette agli utenti di estrarre un unico modello di dati dalle numerose e disparate informazioni e sorgenti di content, di accedevi e di manipolarle come se risiedessero in un'unica locazione.

La scelta di uno, dell'altro o di entrambi i software è regolata dallo stile di programmazione desiderato e da una ponderata analisi che contrappone la quantità dei content e quella dei dati a cui si deve accedere. I due tool differiscono tra loro per la tipologia di informazioni a cui accedono e per gli utenti che li utilizzano. DB2 Information Integrator V8.1 è indirizzato a programmatori che hanno familiarità con l'ambiente dei database relazionali e col linguaggio SQL ed è consigliato in progetti che trattano principalmente dati relazionali , documenti XML, o altri applicazioni o tool RDBMS. Al contrario DB2 Information Integrator for Content è rivolto al segmento del content management; oltre alla ricerca federata, il software, offre anche sofisticate procedure di information mining, al fine di individuare nuovi metadati presenti all'interno di documenti testuali, e workflow avanzati per facilitare la realizzazione di processi incentrati sui contenuti.

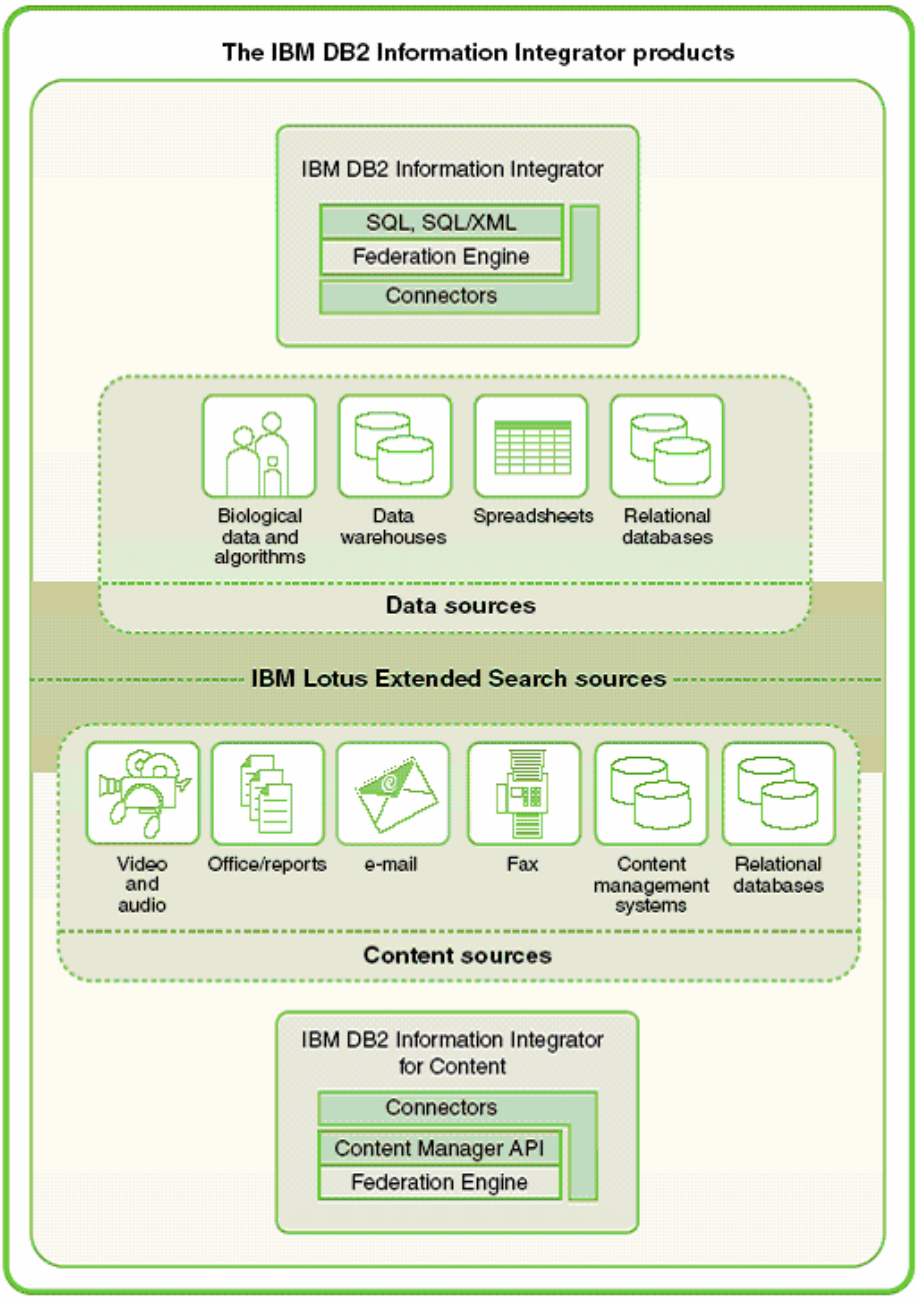



**Figura 3 – Framework strategico d'integrazione dell'informazione per accedere manipolare ed integrare diversi e distribuiti dati in tempo reale**

Il portafoglio prodotti permette di:

- *Scegliere la strategia di accesso ai dati più opportuna*. Riunire i dati rendendoli locali riduce di gran lunga le difficoltà che si incontrano nello sviluppo delle applicazioni fornendo, tra l'altro, migliori performance nell'accesso alle informazioni. È necessario però ricordare che questo approccio introduce problematiche quali i costi dovuti al trasferimento dei dati da una locazione all'altra, all'immagazzinamento degli stessi e alla loro gestione e sincronizzazione. In altri casi potrebbe risultare, invece, più vantaggioso accedere e gestire direttamente i dati nel luogo nel quale si trovano originariamente nei casi in cui vi sia troppa diversità tra di essi, risulti troppo costoso crearne una copia locale o se la loro ubicazione è esterna all'azienda.
- *Integrare dati e content senza trasferirli o senza modificare la piattaforma.* La famiglia DB2 Information Integrator consente l'accesso a dati distribuiti indipendentemente dallo loro collocazione fisica creando un'astrazione che permette all'utente di poterli considerare come provenienti da una singola locazione. Questo permette l'accesso ad un vasto range di sorgenti di dati a cui si può accedere out-of-the-box includendo anche informazioni non strutturate sia interne che esterne all'impresa. Con più informazioni disponibili reperite con metodi più semplici si ha l'opportunità di ottenere un maggior profitto dall'assetto informativo esistente.
- *Raggiungere più velocemente e a un minor costo progressi maggiori.* Attraverso i prodotti della famiglia DB2 Information Integrator è possibile sviluppare con maggior semplicità e rapidità una nuova generazione di applicazioni diversificate che richiedono un'efficiente integrazione dei dati distribuiti; gli sviluppatori possono scegliere indifferentemente se usare un modello di programmazione SQL o DB2 Content Manager. Rendendo possibile l'integrazione nonché la combinazione di dati relazionali e di dati non strutturati DB2 Information Integrator accelera lo sviluppo dei progetti, incrementa le abilità già esistenti e permette la riduzione dei costi di manutenzione.

## 2.1 DB2 Information Integrator V8.1

DB2 Information Integrator V8.1 è studiato per gli sviluppatori di applicazioni in possesso di una certa famigliarità con gli ambienti relazionali. Il software trova le sue radici nella tecnologia DB2 e rappresenta il passo successivo a una serie di prodotti realizzati dall'IBM come IBM DB2 DataJoiner, IBM DB2 Relational Connect e IBM DiscoveryLink.
DB2 Information Integrator fornisce la capacità di federare, ricercare, memorizzare, trasformare e riprodurre i dati; le applicazioni che utilizzano o i tool che generano SQL possono, tramite questo programma, accedere e manipolare dati eterogenei e distribuiti attraverso un federated data server. Attraverso esso, DB2 Information Integrator V8.1 permette l'accesso a DB2 Universal Database, DB2 Informix Edition products, a database Microsoft, Oracol, Sybase e Teradata e a dati semi-strutturati appartenenti a documenti XML, Microsoft Excel, flat file, sorgenti ODBC (open database connectivity) o OLE DB, Web services, messaggi WebSphere MQ e altri ancora.

DB2 Information Integrator ha subito un processo di ottimizzazione che ha ulteriormente sviluppato e potenziato le sue caratteristiche; le principali vengono qui di seguito riportate.

- Query rewrite: fase fondamentale dell'ottimizzazione della query col compito di trasformare interrogazioni non accuratamente elaborare in forme semanticamente equivalenti allo scopo di incrementare le performance limitando o favorendo particolari trasformazioni in base alla loro effettiva applicabilità alle sorgenti di dati di riferimento.
- Pushdown analysis: fase di nuova introduzione che determina quanto di ogni singola query debba essere esaminato da ogni singolo server terminale e quante elaborazioni compensative hanno bisogno di ricorrere al sistema DB2 Information Integrator.
- Cost-based optimization: crea un piano esecutivo d'interrogazioni basato sui costi stimati i quali includono, tra gli altri, statistiche standard dalle sorgenti di dati (ad esempio cardinalità e indici), capacità dei server dati, dell'I/O e della rete.

- Statement generation: stadio che genera un piano eseguibile realizzato in relazione all'ottimizzazione dei costi illustrata precedentemente.
- The query run-time engine: instrada l'esecuzione di una query attraverso i vari dati con cui essa deve interagire, siano essi locali o remoti, fornendo una vista coerente di un database virtuale.

### 2.1.1 Versioni e Sistemi Operativi

DB2 Information Integrator V8.1 è compatibile con diversi sistemi operativi:

- AIX
  - AIX (32-bit) 4.3.3
  - AIX (32-bit e 64-bit) 5.1.0
  - AIX (32-bit e 64-bit) 5.2.0
- HP-UX 11i e sucessive (sia 32-bit che 64-bit)
- Intel Linux, sistemi AMD
- Solaris 7,8 e 9 (32-bit e 64-bit)
- Windows
  - -Windows NT 4.0 con SP 6a e successivi
  - -Windows 2000 Professional
  - -Windows XP 32-bit (supportato solo per funzioni di test e di sviluppo)

Il software viene proposto in cinque edizioni:

- *DB2 Information Integrator Replication Edition:* è stata realizzata per la riproduzione dei dati necessaria alle grandi e medie imprese. Questa edizione rappresenta una soluzione d'integrazione ideale in situazioni che richiedano la copia di dati relazionali al fine di incrementare le prestazioni e l'accessibilità delle query. Le possibilità di management e di amministrazione integrate in DB2 Information Integrator Replication Edition permettono una gestione delle copie end-to-end fra le risorse relazionali più comuni come DB2 Universal Database, Informix Dynamic Server, Microsoft SQL server, Oracle ed altri. Questa edizione è compatibile con i sistemi operativi Unix, Windows e Intel Linux ed è concessa in licenza in base al numero di processori e di sorgenti di dati collegate.
- *DB2 Information Integrator Standard Edition:* offre, in un singolo pacchetto, sia funzioni di replicazione che capacità di accesso federato. Con questa edizione le aziende hanno la possibilità di includere nelle proprie applicazioni dati in tempo reale, semi-strutturati, non relazionali o content rilevanti ottenendo, così, risultati più aggiornati e maggiormente dettagliati; le imprese, inoltre, possono ridurre i costi e creare progetti d'integrazione più velocemente utilizzando gli ambienti di sviluppo standard. DB2 Information Integrator Standard Edition è compatibile con gli stessi sistemi operativi elencati al punto precedente ed è concesso in licenza secondo i medesimi criteri.
- *DB2 Information Integrator Advanced Editions:* fornisce tutte le funzionalità dell'edizione precedentemente illustrata arricchendole con la possibilità di ricorrere a quelle messe a disposizione da DB2 Universal Database Enterprise Server Edition (ESE) come le capacità di gestione di database complessi e le funzioni d'integrazione dell'informazione che, insieme, rappresentano una solida piattaforma per le necessità d'information management. Anche DB2 Information Integrator Advanced Edition può essere utilizzata con i sistemi operativi Unix, Windows e Intel Linux ed è concessa in licenza in base al numero di processori e di sorgenti di dati collegate.
- *DB2 Information Integrator Advanced Editions Unlimited:* rappresenta un'ulteriore miglioramento della versione precedente conservandone tutti i punti di forza eliminando la necessità di acquisire le licenze di DB2 Information Integrator Connector. Mentre i sistemi operativi compatibili rimangono gli stessi delle altre edizioni la licenza viene concessa solo in base al numero di processori.

- *DB2 Information Integrator Developer Edition:* si presenta come un pacchetto a basso costo indirizzato a sviluppatori di singole applicazioni orientato a guidarli durante la progettazione, la realizzazione e il collaudo delle stesse. Questa ampia e completa offerta comprende, oltre a tutte le funzionalità di DB2 Information Integrator Advanced Edition, anche toolkit che permettono di estendere l'accesso a sorgenti di dati personalizzate, WebSphere e tool di progettazione DB2. Il software presente in questo pacchetto è concesso in base al numero di utenti.

## 2.1.2 Struttura del tool

La piattaforma di Integrator Informator utilizza, oltre al federated data server in parte già descritto, un replication server e un database server garantendo una flessibilità eccezionale nella gestione dei cambiamenti dei carichi di lavoro grazie alla fusione dei punti di forza di questi componenti.

Il *Federate data server*, figura 4, fornisce alle applicazioni la caratteristica del "tempo reale" e un accesso integrato a diverse sorgenti come se esse fossero una sola indipendentemente dal formato, dalla locazione e dal sistema operativo. Inoltre assicura tutte quelle caratteristiche di trasparenza, eterogeneità, estensibilità ecc. descritte nel capitolo precedente (vedere 1.1.1).

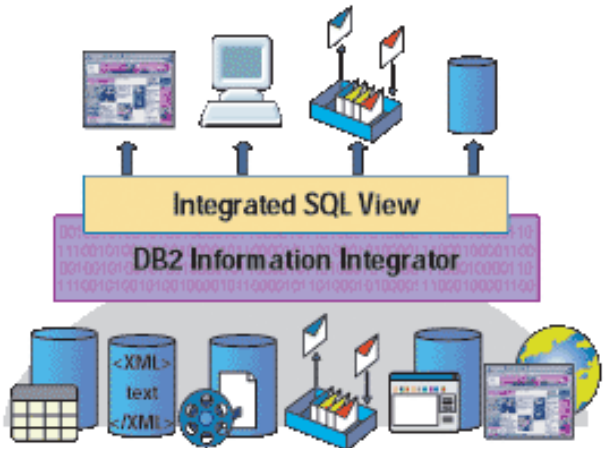



**Figura 4 – Federated data server**

Il *Replication server* consente di sfruttare in maniera ottimale il data placement al fine di ottenere alti livelli di prestazioni e di disponibilità; gli utenti, inoltre, possono configurare una vasta varietà di caratteristiche inerenti alla topologia, alla latenza e alla consistenza. Viene infatti fornita loro la possibilità di collegare database relazionali differenti come quelli della famiglia IBM (DB2 Universal Database e DB2 Informix Dynamic Server), di Microsoft, di Oracle e di Sybase.
Questo server supporta sia quella che letteralmente viene chiamata distribution e che rappresenta il trasferimento di informazioni da un database verso molti, sia la consolidation ovvero il procedimento opposto che fa convergere in un'unica locazione informazioni recuperate da diverse sorgenti. Le trasformazioni possono essere applicate, attraverso espressioni SQL standard o l'esecuzione di stored procedure, contemporaneamente al trasferimento dei dati, il quale può essere automatizzato da una programmazione d'esecuzione a intervalli predefiniti, ininterrotta o guidata dal verificarsi di determinati eventi.

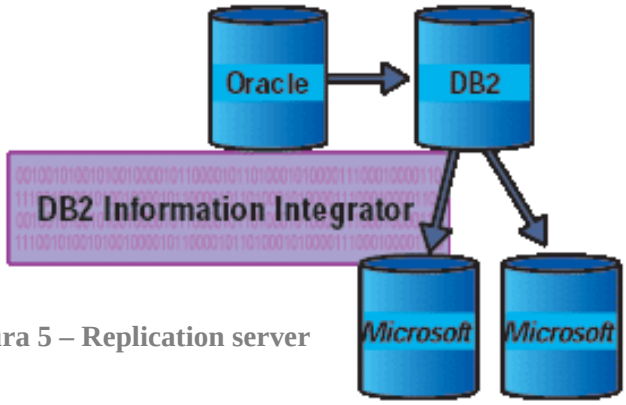



**Figura 5 – Replication server**

Il replication server è comunemente sfruttato per mantenere i data warehouse o i marts dei sistemi societari centrali in sincronia con le filiali o per garantire l'operatività dei data stores.

Il *Database server*, DB2 Universal Database, permette la gestione integrata e la memorizzazione dei metadati. Le versioni DB2 Information Integrator Advanced Edition e Advanced Edition Unlimited rendono DB2 UDB utilizzabile direttamente dalle applicazioni.

## 2.1.3 Punti di forza del software

Molte sono le aree in cui DB2 Information Integrator ha introdotto significativi vantaggi rispetto alle tecnologie già esistenti, qui di seguito vengono elencate quelle ritenute maggiormente rilevanti.
*Prestazioni*. Non essendoci industrie di riferimento che propongono software che permettono l'accesso federato ai dati, l'IBM ha confrontato il proprio prodotto con DB2 DataJoiner mettendo in luce che per query complesse, relative a molte sorgenti, DB2 Information Integrator risulta più veloce rispetto a DB2 DataJoiner nell'80% dei casi, nella metà dei quali ha mostrato addirittura una velocità di risposta pari al doppio di quella del software col quale è stato comparato.
*Usabilità*. L'amministrazione dei processi di federazione e di replicazione è integrata all'interno di un centro di controllo. Wizards aiuta chi per la prima volta interagisce col tool a districarsi nei vari passaggi di configurazione permettendo, tra l'altro, il frazionamento della procedura d'amministrazione in una serie di compiti più elementari al fine di ottenere una più efficiente manutenzione del sistema.
*Sorgenti supportate*. DB2 Information Integrator estende significativamente l'insieme di sorgenti a cui si può accedere, in particolar modo consente l'interazione con complessi documenti, content repositories, sistemi di posta elettronica e Web.
*Capacità d'interrogazione*. Gli utenti di DB2 DataJoiner potranno apprezzare il significativo arricchimento delle capacità d'interrogazione in quanto, mentre questo software si appoggiava a DB2 V.2 SQL, DB2 Information Integrator si avvale di DB2 V.8 SQL.

La seguente tabella mostra le funzionalità disponibili con DB2 Information Integrator confrontate con DB2 Data Joiner, DB2 Relational Connect, DB2 Life Sciences Data Connect e DB2 V8.1

| **Funzione** | **DataJoiner** | **DB2 UDB 7.2 & RL/LSCD** | **DB2 UDB 8.1** | **DB2 II 8.1** |
|---|---|---|---|---|
| DB2 Informix Lettura/Scrittura (R/W) | ✓ | ✓(R/O) | ✓(1PC) | ✓ |
| Oracle, Sybase, SQL Server, Teradata R/W | ✓ | ✓ (R/O) | | ✓ |
| ODBC R/W | | | | ✓ |
| Excel, flat file, Life Sciences wrapper (R/O) | | ✓ | | ✓ |
| Wrapper XML (R/O) | | ✓ | | ✓ |
| Wrapper di ricerca estesa (R/O) | | | | ✓ |
| MQ Series (1PC) | | ✓ | ✓ | ✓ |
| Wrapper Architecture | | ✓ | | ✓ |

| | | | | |
|---|---|---|---|---|
| Toolkit wrapper per programmatori | | | | ✓ |
| Centro di controllo – Rilevazione dei nickname | | | | ✓ |
| Caching - MQTs over nickname | | | ✓ | ✓ |
| Web services provider | | ✓ | ✓ | ✓ |
| Web services consumer | | | ✓ | ✓ |
| Sviluppo di applicazioni attraverso tool standard (WS Studio, MS Visual studio) | | ✓ | ✓ | ✓ |
| Replicazione di dati eterogenei | ✓ | | | ✓ |

**Tabella T1 – Funzionalità di DB2 Information Integrator a confronto con altri prodotti di casa IBM**

## 2.1.4 Utenti

DB2 Information Integrator è concepito per essere utilizzato da tre tipologie di utenti: quelli finali, gli amministratori e i programmatori.
Nella sua azione di supporto a quest'ultima categoria, il software fa da ponte fra i componenti e la logica aziendale risiedenti nel secondo livello dell'architettura e-business e le sorgenti di dati permettendo, così, ai programmatori di interagire con svariati ed eterogenei database, data management system e file come se si trattasse di un unico e immenso database aziendale.
Per quanto riguarda le funzionalità messe a disposizione agli amministratori si possono citare gli strumenti che permettono di mantenere ed implementare la sicurezza del federated data server e che verranno approfonditi nel prossimo paragrafo.
Per agevolare gli utenti finali, il programma permette la realizzazione di transazioni SQL includendo sia le istruzioni elementari come add, update, remove sia l'utilizzo di query complesse che accedono ad un vasto range di sorgenti di dati remote sia strutturate che non.

### 2.1.4.1 Il punto di vista dei programmatori e dei database administrator

Per i programmatori DB2 Information Integrator semplifica notevolmente ambienti molto complessi. Attraverso una singola query SQL, il programmatore medio ha la possibilità di accedere a dati sia attraverso svariati database che mediante sorgenti non relazionali.
Il tool, inoltre, non si limita a ridurre la complessità dei programmi eliminando la necessità di conoscere diverse versioni SQL, gestendo le connessioni simultanee tra diversi database e amministrando complessi join logici necessari per mettere in relazione sorgenti distinte, bensì abbassa il livello di competenze e capacità che il programmatore stesso deve possedere per districarsi all'interno di un ambiente IT eterogeneo.
Il compito che spetta ai database administrator (DBA) è, appunto, quello di configurare un ambiente che permetta tutto questo. Attraverso il DB2 Control Center, il DBA definisce gli attributi relativi alle sorgenti di dati critiche ovvero specifica il tipo di sorgente, i server sui quali esse risiedono, la mappatura degli utenti e dei campi delle entità in gioco nello schema relazionale, che prendono il nome di nickname. In molti casi il DBA può stabilire anche semplici trasformazioni che permettono ai dati risiedenti in un determinato database di essere elaborati coerentemente con quelli presenti su di un'altra base di dati; per esempio, potrebbe indicare che gli codici "M" ed "F" relativi al sesso di un individuo sostituiscano i valori "1" e "2" di un'altra sorgente. Una volta terminata la configurazione dei nickname, l'indirizzamento dei server e l'eventuale dichiarazione delle trasformazioni descritte in precedenza, il programmatore non trova davanti a sé nessun tipo di ostacolo e può agevolmente utilizzare DB2 Information Integrator.

## 2.2 DB2 Information Integrator for Content

DB2 Information Integrator for Content V8.2 è finalizzato alla gestione di dati non strutturati ed è stato progettato per incrementare le performance e consentire l'accesso anche a informazioni critiche (sia strutturate che non) presenti sia all'interno che all'esterno dell'azienda. Esso agevola il compito delle imprese di incontrare i cambiamenti del business collocando informazioni e contenuti necessari all'azienda in un'aperta ed estensibile struttura d'integrazione dell'informazione.
Questo tool, rivolto principalmente agli sviluppatori di content applications, presenta le stesse funzionalità offerte dal suo predecessore IBM Enterprise Information Portal aggiungendo, ad esse, operazioni EII maggiormente orientate verso l'integrazione dei contenuti e basate sul modello di programmazione IBM Content Management.

### 2.2.1 Struttura del tool

Il tool si compone degli elementi base mostrati in figura 6:

- *Un Federated server*, il quale consente l'accesso distribuito a diverse sorgenti dati e di content. La query federata include testo intelligente e richieste immagine attraverso sorgenti di content dedicate.
- *Connettori* a una vasta gamma di tipologie di sorgenti di dati e di content che permettono anche ad altri elementi di accedervi in un modo completamente trasparente.
- *Un server di ricerca estesa* per elaborare query federate sia attraverso content server dedicati che attraverso internet.
- *Un toolkit di estrazione dell'informazione* per catalogare ed indicizzare documenti a cui si accede direttamente dai content servers di appartenenza
- *Un applicazione workflow avanzata* per gestire il ciclo di vita dei vari content

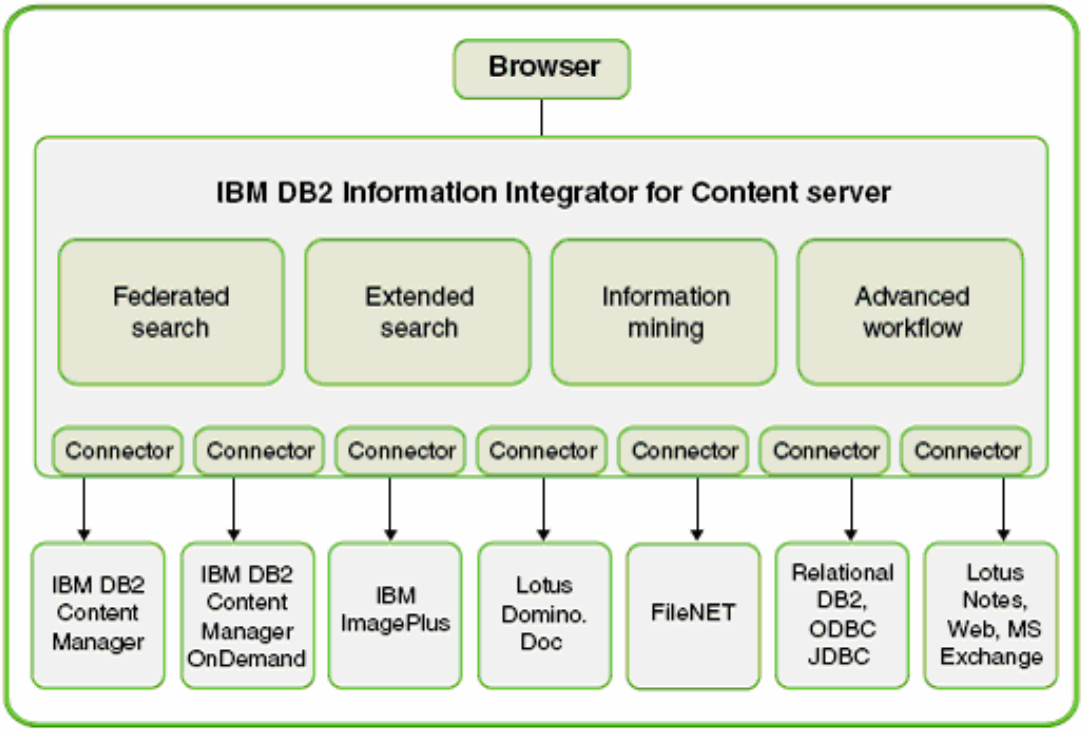



**Figura 6 – DB2 Information Integrator for Content**

### 2.2.1.1 Il federated server e i connettori

Con lo scopo di accelerare le operazioni di implementazione dei progetti di content integration, questo software consente di accedere simultaneamente, attraverso una singola ricerca, a varie sorgenti di dati out of the box nelle quali gli attributi locali sono mappati in un insieme di attributi definiti a livello globale.
Ricorrendo ad una libreria di connettori che mettono in comunicazione i livelli applicativi del server DB2 Information Integrator for Content e dei content servers remoti, permette l'accesso e l'interazione coi seguenti ambienti:

- La famiglia IBM DB2 Content Manager, che comprende DB2 Content Manager, DB2 Content Manager OnDemand e IBM ImagePlus
- Database Lotus, come ad esempio IBM Lotus Domino.Doc
- Ogni database relazionale, come DB2 o Oracle, che è compatibile ODBC JDBC (Java database connectivity )
- Archivi di content, come FileNet Panagon Information Services per esempio
- Lotus e Microsoft e-mail e sistemi di collaborazione tipo lightweight LDAP (directory access protocol) directory e più di 18 siti di ricerca grazie alla presenza di un connettore a IBM Lotus Extended Search.

### 2.2.1.2 Il server di ricerca estesa:Lotus Extended Search

Un utente, attraverso un singolo punto d'accesso, può ricercare e quindi ottenere informazioni sia attraverso intranet che per mezzo di internet. Lotus Extended Search rende possibile l'esecuzione di ricerche federate senza esigere la creazione o la conservazione di un indice centrale per l'accesso a content risiedenti in database multipli, applicazioni, sorgenti Web o indici. Gli utenti possono eseguire semplici query che saranno automaticamente tradotte, dal motore di ricerca avanzata, nel linguaggio d'interrogazione originale della sorgente di dati interessata. Utilizzando i metodi di ricerca e di recupero propri delle sorgenti, il software elimina la necessità, per l'utente, di padroneggiare le complessità della sorgente dati principale; in più permette la ricerca parallela di informazioni attraverso differenti sorgenti dati realizzando, così, ricerche più veloci e maggiormente efficienti.
Appoggiandosi a Lotus Extended Search è possibile realizzare queste ricerche attraverso interfacce di facile utilizzo; gli utenti ricevono, nell'istante in cui vengono formulati, i risultati delle ricerche sottoforma di collegamenti ai documenti coinvolti senza la necessità di aspettare i tempi di risposta delle sorgenti di dati remote.
È possibile:

- Ricercare e recuperare documenti da e-mail, collaboration servers sia in ambiente Lotus che Microsoft
- Effettuare ricerche attraverso domini come Google o Yahoo! e siti d'informazioni come CNN o BusinessWire
- Ricercare file system, LDPA directory e database relazionali

Qualsiasi sia la tecnica di ricerca adottata occorre attuare provvedimenti che garantiscano il mantenimento della sicurezza dei dati; grazie alle capacità federali di ricerca che DB2 Information Integrator for Content mette a disposizione tale requisito viene tutelato.

### 2.2.1.3 Estrazione delle informazioni

Per sfruttare al meglio i potenziali benefici che tutte le informazioni possedute da un'impresa possono offrire, DB2 Information Integrator for Content permette di indicizzare, schematizzare e catalogare i content necessari alla società avvalendosi di Web crawlers i quali scansionano l'intranet aziendale, extranet e file system locali e che possono, tra l'altro, interagire con database IBM Lotus Note e IBM Lotus Domino o altre sorgenti di content.

L'estrazione dell'informazioni, anche chiamata text mining, raccoglie e riassume le informazioni inerenti a tutti i content disponibili nel federated server con l'ausilio dei connettori; utilizzando il Web crawling e algoritmi di estrazione testuale, ha la possibilità di rendere strutturati content che altrimenti non lo sarebbero. Tali algoritmi permettono di identificare il linguaggio in cui sono scritti i documenti, individuare le loro caratteristiche interne (come ad esempio i nomi), classificarli conformemente alla tassonomia definita, raggrupparli in base alla categoria di appartenenza e riassumerli creando, in questo modo, metadati supplementari per gli oggetti analizzati.
Le caratteristiche fondamentali di questo processo riguardano:

- L'identificazione del linguaggio nel quale il documento è scritto
- La catalogazione del documento in base a classi preesistenti suddivise in base ad una tassonomia predefinita (come linee di prodotti o concorrenti)
- Il reperimento di informazioni con lo scopo di identificare le principali entità all'interno del documento (come nome dell'organizzazione, termini di dominio tecnico, abbreviazioni, date ecc.)
- La sintetizzazione dell'estrazione delle frasi più importanti da ogni documento al fine di creare prospetti
- Gruppi di raccolta per mettere automaticamente in relazione documenti in base al loro contenuto senza dover ricorre a classi di catalogazione predefinite.

La Gartner Inc. ha affermato: "…Così come un'azienda è a conoscenza del fatto che deve impegnarsi per ottenere i massimi benefici dai propri sforzi, essa ha anche compreso che nel perseguire questo scopo è fondamentale investire nella costruzione di una classificazione aziendale. Esiste una probabilità dello 0.7% che entro il 2005 le imprese stanzieranno minino il 15% del loro budget di content management nella creazione della tassonomia e negli strumenti o nelle funzioni di metatag per trarre il massimo profitto dagli investimenti inerenti ai contenuti…"[2].

### *2.2.1.4 Implementazione di un workflow in un ambiente federato*

Ultima caratteristica fondamentale di DB2 Information Integrator for Content è quella di consentire l'inserimento delle informazioni all'interno di processi di workflow. Questo può essere facilmente realizzato attraverso l'utilizzo di un software grafico che permette una più agevole definizione del workflow ottimale per l'azienda. Essa può ottenere eccellenti vantaggi competitivi ricorrendo alla ridefinizione di processi lavorativi convenzionali con l'ausilio della digitalizzazione dei dati; si pensi solo ai benefici che può portare l'adozione di un Web self-service, o di altre iniziative a supporto della clientela, a fianco del tradizionale call center.
Attraverso questa riorganizzazione dei workflow incentrata sull'informatizzazione è possibile incrementare la produttività dell'impresa e il livello di riutilizzazione dei content, ridurre i tempi di produzione, migliorare la conoscenza dei clienti assicurandosi, così, la loro fedeltà e la loro soddisfazione, affinare collaborazioni e comunicazioni tra i diversi reparti e conquistarsi rapidamente le nuove opportunità offerte dal mercato con tutti i ben noti benefici che ciò comporta

[2] Gartner, Inc., Seven Areas of Content Management Growth for 2003, 13 Novembre 2002, M. Gilbert, D. Longan and K. Shegda.

## 2.3 DB2 Information Integrator Software Services

Per completare il quadro dei prodotti della famiglia DB2 Information Integrator è, qui di seguito, riportata una semplice descrizione dei servizi software ad essa associati.
Gli IBM Software Service per DB2 Information Management primeggiano nella grande impresa, nelle soluzioni end-to-end con un'attenzione particolare al successo a lungo termine del cliente. Essi sono realizzati al fine di sfruttare al meglio l'utilizzo della tecnologia DB2 Information Integrator per dar vita a opportunità di mercato competitive salvaguardando gli investimenti nell'information technology e contemporaneamente ottenendo prestazioni soddisfacenti dall'intero sistema.
Gli IBM Global Service, invece, rappresentano un potente strumento per la progettazione, lo sviluppo, e il supporto di procedure di riproduzione di database.
Questi tool cooperano al fine di fornire agli utenti le conoscenze necessarie per gestire tutti gli aspetti dell'information management mettendo a disposizione un'ineguagliabile competenza nella modellazione delle informazioni e nel supporto generale attuando azioni di supporto per l'installazione, la configurazione, e la manutenzione di tutti i prodotti della famiglia IBM DB2 Information Integrator.

| ***SERVIZIO*** | ***DESCRIZIONE*** |
|---|---|
| ***Information Integrator V8.1*** | |
| IBM Design and Planning Services for DB2 Information Integrator | Interpreta le richieste d'integrazione dei dati e progetta una soluzione che include una chiara rappresentazione dei dati federati. |
| IBM Implementation Services for DB2 Information Integrator | Comprende i processi di pianificazione, d'installazione, di migrazione, di configurazione e d'integrazione di DB2 Information Integrator per gli ambienti AIX, Windows e OS/390. Questi servizi permettono di riprodurre, collegare e integrare diverse sorgenti di dati in modo da realizzare una più efficiente via d'accesso, d'immagazzinamento ed estrazione delle informazioni all'interno dell'azienda. |
| IBM Fast Services for DB2 Information Integrator | Rappresenta un ausilio per coloro che, per la prima volta, iniziano ad utilizzare DB2 Information Integrator |
| IBM Performance Turing Services for DB2 Information Integrator | Analizza le prestazioni del software individuando i cosiddetti "colli di bottiglia" e possibili ottimizzazioni del sistema. |
| ***DB2 Information Integrator for Content*** | |
| Content Manager for Multiplatform Basic Installation & Configuration | Servizi raccomandati per DB2 Content Manager per l'installazione e la configurazione di multipiattaforme (può includere DB2 Information Integrator for Content) |
| Solution Integration | Permette una profonda conoscenza del prodotto e delle abilità necessarie per la progettazione e la implementazione di soluzioni integrate per tutti i prodotti di Content Manager |

**Tabella T2 – DB2 Information Integrator Software Services**

# Apporto di DB2 Information Integrator nelle data warehouse

## 3.1 Relazione fra l'integrazione dell'informazione e i data warehouse

L'odierna architettura stratificata dei data warehouse (di cui viene mostrato lo schema a blocchi in figura 7) è basata sul presupposto che tutti i dati richiesti per una particolare query o report debbano essere trasferiti in un singolo data mart o, per lo meno, in un singolo ambiente di data warehouse grazie all'utilizzo delle funzioni ETL, questo allo scopo di garantire stabilità, consistenza delle informazioni ed un sicuro accesso ad esse.

Le sempre più incessanti necessità di riduzione della latenza, di diminuzione della permanenza di dati locali poco utilizzati, nonché la diversità delle sorgenti a cui si deve accedere sono problematiche che trovano soluzione nell'utilizzo di query distribuite che, attraverso le funzionalità federali, permettono di poter considerare un singolo warehouse o mart virtuale senza il trasferimento fisico descritto in precedenza.
È comunque necessario sottolineare che questo approccio in nessun caso potrà sormontare quello tradizionale il quale non verrà mai completamente abbandonato a causa dei ben noti problemi di performance, inconsistenza ed autonomia, ma potrà essere usato per migliorare, potenziare ed ottimizzare gli esistenti data warehouse orientandoli verso le specifiche necessità aziendali ovviamente solo nei casi in cui si possa trarre reale beneficio dall'utilizzo di questo metodo.

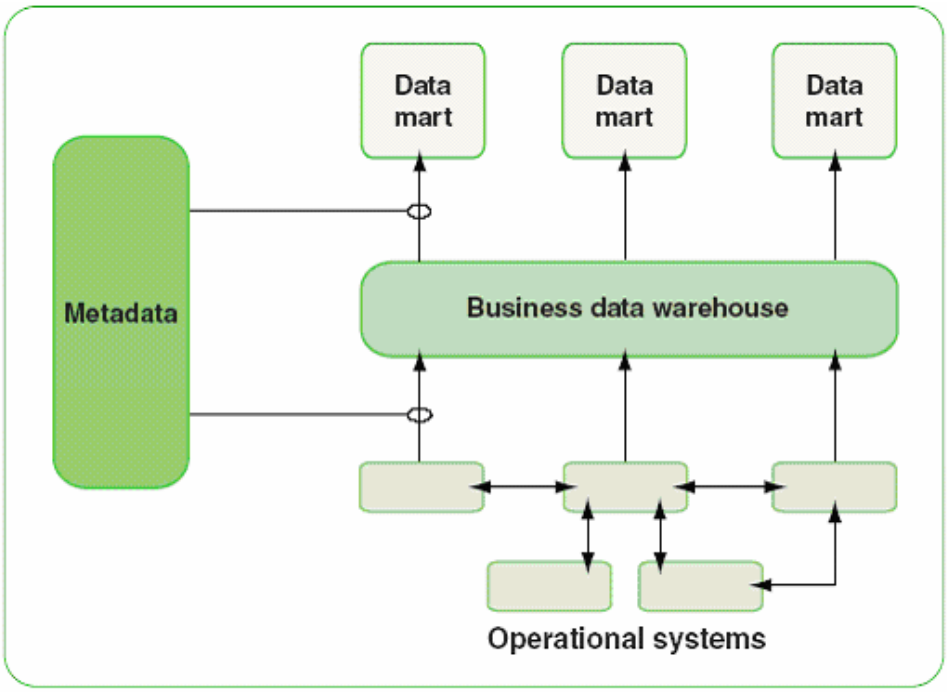



**Figura 7 – Struttura di un data warehouse a tre strati, architettura tipica dell'ambiente IBM**

## 3.2 Accesso a dati in tempo reale

L'approccio federale può rivestire un importante ruolo in ambienti in cui sia necessario accedere a specifici elementi in tempo reale e, allo stesso tempo, possedere informazioni riguardanti dati storici o analitici già presenti all'interno del data warehouse. Questo scenario è mostrato in figura 8.

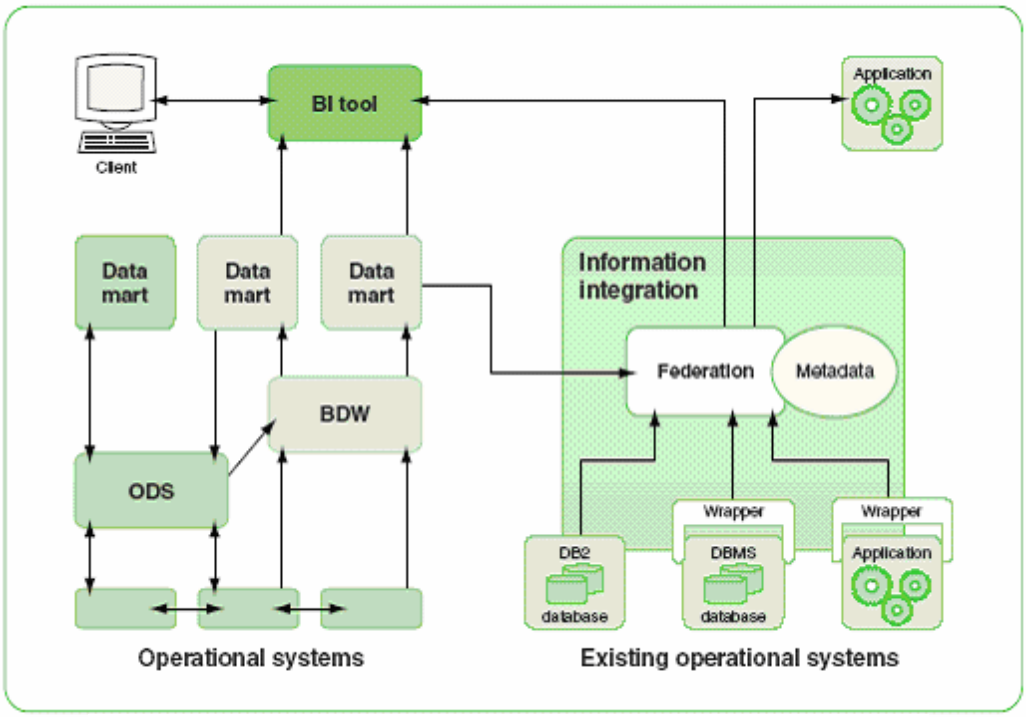



**Figura 8 – Accesso federato a dati real time**

Consideriamo una query in gran parte imperniata su dati storici o consolidati disponibili nel data mart che però richiede anche il reperimento di alcune informazione maggiormente aggiornate. In un'architettura tradizionale questi dati dovrebbero essere inseriti continuamente nel data mart, solitamente tramite un ODS[3], il che implicherebbe non solo trasferimenti e immagazzinamenti significantemente voluminosi, ma un ambiente ETL capace di mantenere un throughput vicino al tempo reale. Federation offre una più semplice ed elegante soluzione in molte di queste situazioni. Quando una query è in esecuzione è possibile inviare al sistema operazionale una semplice richiesta per uno specifico frammento d'informazione, il risultato restituito viene collegato con l'informazione recuperata dal mart; questo elimina la necessità di immagazzinare i dati in tempi vicini a quelli reali o di assegnare all'ambiente ETL il gravoso compito di gestione di tali informazioni. Le query trasmesse al sistema operazionale dovrebbero essere semplici o comunque in un formato con il quale esso riesca ad interagire efficacemente e avrebbero il compito di restituire un ristretto range di specifici frammenti d'informazione il che limita pesantemente il livello di prestazione. Le query federate adottano lo standard SQL permettendo, tra l'altro, un uso trasparente affiancato dagli esistenti analitici business intelligent (BI) tool che, in questo modo, possono accedere a dati relazionali, e non, sia a livello locale che a livello remoto. Questo salvaguarda gli investimenti aziendali attuati negli applicativi già adottati e influisce sulle capacità e le competenze degli sviluppatori IT nell'utilizzo di questi tool e dei loro paradigmi SQL.
Federation non si limita a fornire un accesso ai dati in tempo reale ma può essere utilizzato per recuperare ogni tipo d'informazione senza fare ricorso a procedure di memorizzazione. Come ben noto i dati risiedenti in un data warehouse appartengono a quell'insieme di informazioni di cui si potrebbe aver bisogno in un determinato momento, nonostante questo, però, è importante sottolineare che in molte architetture la gran parte di essi, dal 20 al 50%, vengono raramente utilizzati. Quando l'accesso a determinati dati avviene con saltuarietà e quando essi risiedono anche in altre locazioni è utile ricorrere all'approccio federato in modo da reperire le informazioni

[3] Un ODS può essere visto sia come uno strato addizionale tra i sistemi operazionali e il BDW (business data warehouse), implicando il passaggio di tutti i dati attraverso l'ODS, sia come forma di bypass del BDW stesso.

direttamente dalle sorgenti remote in cui sono memorizzate; per dati storici, al contrario, è consigliabile mantenere una copia all'interno del data warehouse in quanto l'unico duplicato presente si trova su dischi magnetici di backup. Esiste un ulteriore beneficio dato dal ricorso all'infrastruttura federale caratterizzato dalla possibilità delle imprese di realizzare applicazioni operazionali che riescano, in modo semplice ed immediato, ad accedere ai dati presenti nel data warehouse ed a combinarli con le informazioni operazionali esistenti nelle sorgenti remote così come mostrato in figura 8.

## 3.3 Accesso a contenuti non strutturati

La figura 9 mostra un'altra situazione nella quale federation può coadiuvare il data warehouse: le esigenze aziendali, in questa circostanza, riguardano l'acquisizione di dati non strutturati o di contenuti da inserire in report generati e risiedenti all'interno dell'ambiente locale. Nelle architetture tradizionali i passi da compiere risulterebbero essere due, il primo si dovrebbe occupare del reperimento dei contenuti richiesti e del loro inserimento all'interno del data warehouse, il secondo riguarderebbe la fase d'interrogazione delle informazioni trasferite attraverso i consueti procedimenti. La maggior parte di questi dati è molto voluminosa e può far emergere problemi anche per quelle aziende che accettano di immagazzinare grandi quantità d'informazioni, questo perché alcuni contenuti potrebbero essere volatili, su internet o in data store di compagnie partner il che renderebbe difficoltoso conoscere il momento in cui essi perdono di consistenza ovvero il momento in cui subiscono modifiche rendendo necessario il trasferimento dei dati aggiornati. Questa situazione permette di comprendere a pieno la potenza di federation che fornisce l'accesso ai contenuti solo nei tempi e nei modi richiesti attraverso un procedimento che, durante l'esecuzione di un report, invia una sotto-query alla sorgente remota restituendo solo le informazioni aggiornate richieste.

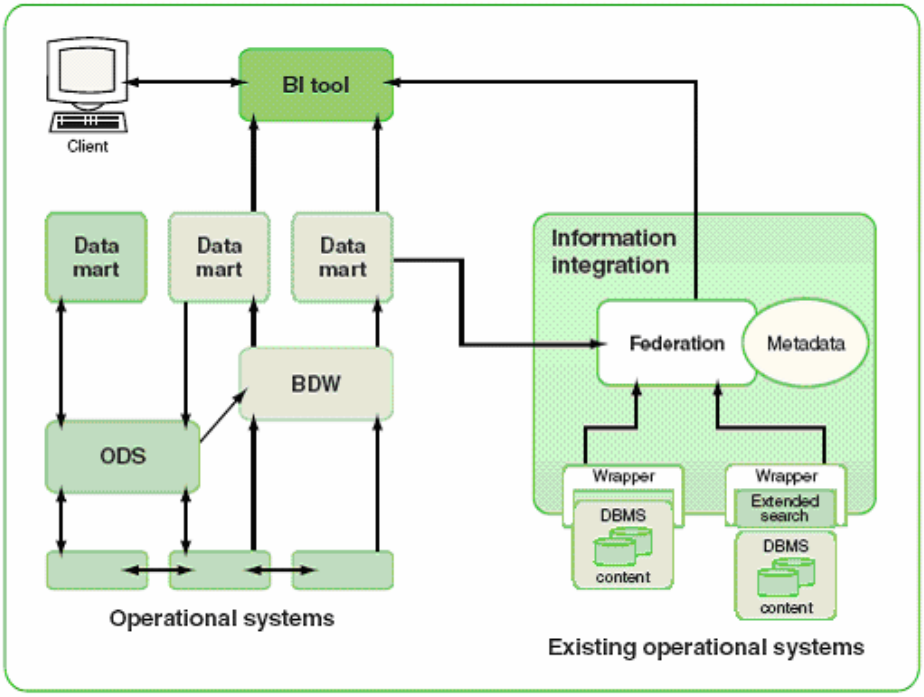



**Figura 9 – Accesso federato a contenuti non strutturati**

## 3.4 Accesso a data mart e business data warehouse

La figura 10 rappresenta una terza possibile estensione del concetto di data warehouse, la quale analizza una situazione ormai diffusa in tutte le aziende moderne che con sempre più frequente regolarità fanno ricorso a più di un data warehouse; questo fenomeno può nascere dalla fusione, dall'acquisizione o semplicemente come risultato di investimenti presi indipendentemente o in maniera non coordinata tra reparti differenti.

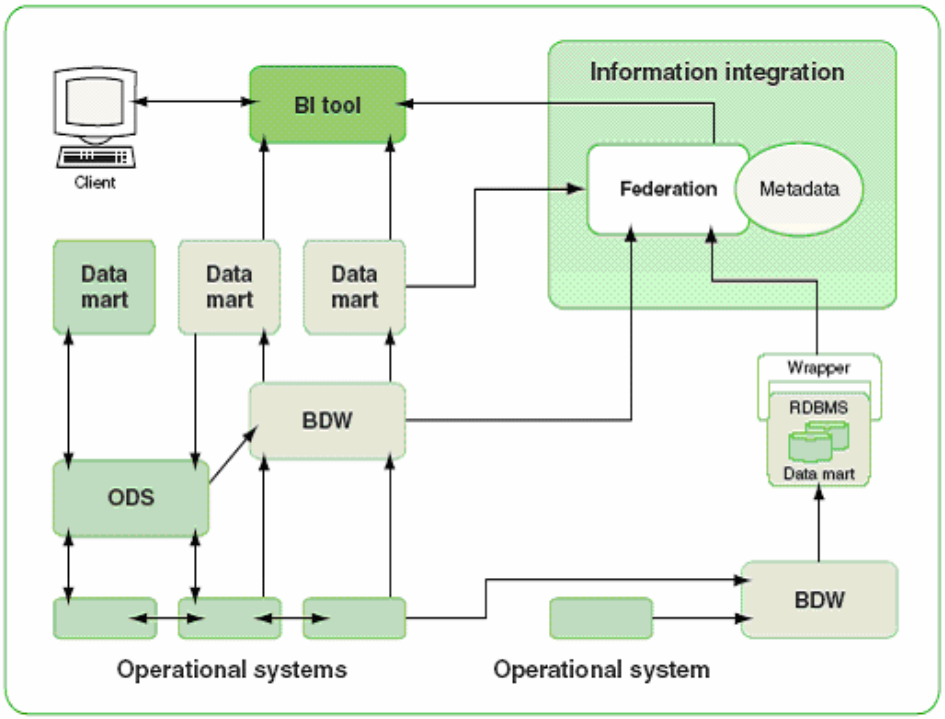



**Figura 10 - Accesso federato a data mart**

In questo contesto sorgono due problemi basilari di non facile soluzione per l'architettura tradizionale del data warehouse. Innanzitutto risulta fondamentale rendere possibile il raggruppamento dei diversi contenuti in un'unica area; questo risultato può essere raggiunto attraverso due strade, o trasferendo il contenuto di un warehouse in un'altro limitandosi agli elementi effettivamente necessari, o realizzando una struttura di immagazzinamento dei dati globale contenente tutte le informazioni provenienti da ogni singolo data warehouse. Totalmente distaccato dalle problematiche relative al volume delle informazioni coinvolte, si trova il secondo dei problemi precedentemente annunciati; esso riguarda la creazione di un modello di dati unificato al fine di convertire ogni sorgente e di permettere, così, alle informazioni presenti su di un data warehouse di poter essere trasferite su di un'altro.

Le soluzioni proposte a queste tematiche risultano, come già detto, di non facile attuazione; l'approccio federale permette, in un modo più agevole, il raggiungimento degli stessi obiettivi utilizzando, però, tecniche differenti. Una query federata, infatti, si rivolge esclusivamente al sottoinsieme di dati necessari per soddisfare le richieste senza dover effettuare alcun tipo di trasferimento evitando, in questo modo, anche la formazione di inutili copie extra.

La figura 10 mostra, inoltre, che è possibile far interagire query federate con uno o più BDW in modo da includere, nell'insieme dei risultati, dati ad un livello di dettaglio estremamente alto che non erano stati precedentemente inseriti nel data mart attraverso il procedimento di popolazione ETL. Deve essere puntualizzato che questo è un approccio che può aumentare infinitamente ampliando il volume della federazione ad ogni step fino ad ottenere, nei casi in cui ne sia presente la necessità, che tutti i dati del mart siano disponibili in uno scenario complessivo e globale. In questo modo le inconsistenze, nel significato o nei contenuti, che inevitabilmente si sono create nel warehouse, durante il suo sviluppo, possono essere scoperte gradualmente e opportunamente

affrontate e risolte. Chiaramente questa funzionalità non è limitata ai casi di data warehouse multipli ma è applicabile anche ad ambienti in cui operano singoli data warehouse consentendo l'accesso occasionale ai dati appartenenti al BDW da parte di utenti di specifici data mart. Infine è, inoltre, possibile decidere se optare o meno per una soluzione che raggruppi fisicamente i mart originari.

## 3.5 Casi in cui conviene ricorrere all'approccio federale

Un potenziale problema riguarda come collegare logicamente e correttamente i dati appartenenti al data warehouse con quelli presenti nei sistemi remoti, tale ostacolo si presenta anche durante il processo di programmazione della fase di popolazione del data warehouse stesso; in entrambe le situazioni è necessario disporre di analisi dettagliate e di una perfetta conoscenza delle sorgenti nonché delle relazioni che le legano con le varie destinazioni rendendo necessario per ambedue un'accurata modellazione dei dati in gioco.
A volte ci si può trovare in situazioni che presentano scenari caratterizzati da relazioni estremamente complesse o da sorgenti la cui qualità risulterà eccessivamente scarsa per consentire un accesso federato, in questi casi se si è a conoscenza del progetto inerente al processo ETL che ha portato alla popolazione del data warehouse è possibile ricorrervi durante la stesura delle query. Si può affermare, dunque, che l'approccio federale non riduce, in nessun caso, la necessità di possedere una completa e dettagliata analisi dell'ambiente in cui si opera, ma può addirittura richiedere maggior rigore nell'esecuzione dei processi a causa della natura inline e in tempo reale di ogni trasformazione.
Tutte queste considerazioni sono alla base delle condizioni al di sotto delle quali il metodo federale può essere utilizzato per estendere il data warehouse, esso rappresenta un potentissimo strumento per soddisfare determinate necessità quali, ad esempio, l'accesso a dati in tempo reale, il reperimento di contenuti il cui trasferimento nel data warehouse risulterebbe proibitivo, il trattamento di dati sporadicamente utilizzati; è inoltre più appropriato per query occasionali rispetto ad interrogazioni ordinarie, con frequenza regolare o delle quali se ne prevede un'assidua ripetizione. Nei casi opposti a quelli appena esposti è, invece, conveniente trasferire i dati nell'ambiente locale.
Non bisogna, però, credere che attraverso federation si ottenga il rimedio a tutti i problemi d'accesso ai dati anche se questo approccio è in grado di risolvere molte delle ben note necessità tipiche di quest'ambito. Come si può ben immaginare il progressivo miglioramento degli strumenti di federazione e la sempre migliore integrazione degli ambienti aziendali permetteranno di aumentare notevolmente le occasioni in cui poter ricorrere a questo metodo; nello specifico ci aspettiamo servizi Web in grado di permettere trasformazioni di dati maggiormente sofisticate consentendo, appunto, l'ampliamento dei possibili impieghi di questi strumenti.

Anche in questo capitolo si è data maggiore importanza alle operazioni di lettura analizzandone approfonditamente metodologie e problematiche; ovviamente, però, l'approccio federale supporta anche operazioni di scrittura che, allo scopo di minimizzare il potenziale impatto negativo sull'integrità dei dati e sulla sicurezza all'interno dell'ambiente operazionale, sono realizzabili solo attraverso le applicazioni responsabili dei processi di creazione e manutenzione dai dati all'interno del sistema stesso.

## 3.6 DB2 Information Integrator e data warehouse

Per comprendere a pieno quanto l'utilizzo di DB2 Information Integrator riesca ad incrementare le prestazioni di un comune data warehouse vengono, qui di seguito, riportati alcuni esempi che ne illustrano tali potenzialità nei differenti ambiti analizzati all'inizio del capitolo.

### 3.6.1 Call center

Consideriamo un call center di una banca o di un altro istituto finanziario nel quale il personale competente ha accesso ad una varietà di informazioni riguardanti i clienti attraverso l'utilizzo di un data warehouse. Tali informazioni potrebbero concernere dettagli sulle transazioni relative ad un certo periodo così come dati derivati o riassuntivi che mostrano trend di comportamento, opportunità di mercato ecc. In questo scenario l'informazione più recente che può essere recuperata riguarda le transazioni effettuate il giorno precedente; la maggior parte dei rapporti di marketing o di trending sono aggiornati con frequenza mensile il che potrebbe portare alla presenza di dati settimanali non aggiornati. Questa breve premessa rende evidente come, in una situazione di questo tipo, un dipendente non sia in grado di rispondere a richieste da parte di clienti che desiderino informazioni riguardo alle transazioni eseguite nel giorno stesso della chiamata e come gli risulti difficile commutare una semplice telefonata in una potenziale opportunità di vendita a causa della carenza di informazioni in tempo reale inerenti a quei clienti che hanno da poco compiuto transazioni bancarie online.
Se l'istituto di credito si appoggia unicamente all'architettura tradizionale del data warehouse sarà in grado di risolvere il primo problema presentato ricorrendo a processi ETL potendo, così, usufruire di dati il più possibile vicino a quelli aggiornati, mentre potrà elidere il secondo attraverso la memorizzazione di vaste quantità di dati supplementari dei quali, per altro, sarà impiegata solo una ristretta porzione per ogni singola richiesta.
L'adozione di DB2 Information Integrator elimina contemporaneamente entrambe le complicazioni esposte attraverso l'approccio già mostrato in figura 8.
Per mezzo di query federate consentirà l'accesso diretto a dati aggiornatissimi attinenti ai clienti indipendentemente dalla loro fisica ubicazione. All'atto di una chiamata da parte di un cliente con perplessità relative alle transazioni odierne il dipendente sarà in grado, attraverso l'esecuzione di una semplice interrogazione SQL, di associare dati presenti nel data warehouse con le informazioni attinenti ai movimenti del giorno stesso recuperate dal sistema branch, da ATM e da canali bancari online.
Il secondo problema è risolto supportando gli impiegati nella fase di presentazione di prodotti finanziari a clienti che hanno contattato l'istituto per altre ragioni, questo è ottenuto dotando i dipendenti stessi di una visione globale più documentata ed esauriente delle proposte che possono avanzare. In questo contesto le informazioni riguardanti gli andamenti o le opportunità contenute nel data warehouse vengono elaborate e presentate assieme ad aggiornati indicatori come, ad esempio, l'estratto conto o la più rilevanti tra le recenti transazioni recuperate dal sistema operazionale.

### 3.6.2 Sistema informativo dirigenziale

L'attuale sistema informativo dirigenziale è largamente, se non totalmente, incentrato sull'obiettivo di fornire dati strutturati ai propri utenti. Tale compito risulta essere molto arduo in quanto una sostanziale quantità di input che partecipa a processi decisionali direttivi si presenta sottoforma di contenuti non strutturati molti dei quali risultano essere esclusi dal sistema informativo dirigenziale a causa delle difficoltà d'accesso, di collegamento con i dati strutturati o dell'eccessivo spazio che occuperebbero se memorizzati nel data warehouse.
DB2 Information Integrator abbatte tutti questi problemi attraverso l'approccio mostrato in figura 9; cooperando con Lotus Exended Search, il tool IBM offre strumenti e modalità d'accesso ad una vasta varietà di sorgenti consentendo la manipolazione simultanea di dati strutturati e di contenuti non strutturati che si preoccuperà di adattare adeguatamente. Usufruendo delle operazioni federate, il data warehouse garantisce un continuo flusso di opportuni contenuti all'interno del sistema informativo senza l'onerosa necessità di doverli memorizzare localmente a meno che ciò non incida positivamente sulle performance dell'intero ambiente.

### 3.6.3 Data mart delle agenzie governative

Come già detto, attualmente la maggior parte delle grandi compagnie fruisce di svariati data warehouse ognuno dei quali interagisce con diversi data mart i quali, a loro volta, rappresentano serbatoi di informazioni conformate alle specifiche necessità dipartimentali e funzionali. Questa situazione è probabilmente più evidente nell'ambiente governativo nel quale diverse agenzie raccolgono e controllano accuratamente la sovrapposizione e, frequentemente, anche i contrasti tra i dati sensibili riguardanti la popolazione, business e risorse. In questo contesto risulta praticamente e politicamente impossibile legare tra loro tutti i data warehouse esistenti in un singolo "mega-warehouse". Tuttavia le odierne esigenze di incrementare la gestibilità, ridurre l'inconsistenza e specialmente potenziare la sicurezza concorrono parallelamente al desiderio di creare, in un modo o nell'altro, una visione unica e globale di tutte queste informazioni. La terza configurazione d'integrazione dell'informazione, illustrata in figura 10, rappresenta il modo per raggiungere tale traguardo.

L'utilizzo di DB2 Information Integrator come "propagatore" di query federate consente alle compagnie di affrontare il problema gradualmente. Esso non necessita di una conoscenza globale della mappatura dei dati in due warehouse, eliminando così anche la necessità di definire un metodo per combinare queste informazioni, bensì parte con una mappatura relativamente ristretta che fornisce valori immediati espandendo, nel tempo, questa soluzione. In questo modo, per esempio, utenti del data mart $\alpha$ possono eseguire una query federata per mettere in relazione un sottoinsieme d'informazioni, appartenenti al data mart $\beta$, con le proprie. Alla fine, ulteriori sottogruppi di informazioni presenti nel data mart $\beta$, possono venire inclusi in query federate facendo, così, crescere la base di metadati e il livello di mapping tra i due mart. Lo stesso procedimento può avvenire anche nella direzione opposta, così come tra altri gruppi di mart all'interno dell'ambiente. Quest'archivio di metadati via via sempre più ampio può, nel tempo, diventare il nucleo per la razionalizzazione dell'archiviazione e della distribuzione di queste informazioni.

# Sperimentazione di DB2 Information Integratior Advanced Edition, Versione 8.1

In questo capitolo verrà trattata tutta la fase di sperimentazione del software, si cercherà quindi di mostrare l'aspetto applicativo delle nozioni teoriche esposte in precedenza.

Parallelamente allo studio del programma è stata portata avanti l'analisi di un altro strumento d'integrazione dell'informazione, questo tool, chiamato MOMIS (Mediator envirOnment for Multiple Information Source), è stato realizzato dal DBGroup dell'Università degli Studi di Modena e Reggio Emilia e verrà descritto nel capitolo successivo. Il fine di questa analisi congiunta è, come si può immaginare, quello di confrontare i due sistemi mettendone in luce pregi e difetti che possono caratterizzare la superiorità di uno o dell'altro prodotto.

## 4.1 Installazione

L'installazione di DB2 Information Integrator risulta essere molto semplice in quanto è supportata dal Wizard che guida l'utente nei vari passaggi necessari per l'ottenimento del risultato finale. In alcuni tratti può, comunque, presentarsi leggermente ostica per utenti meno esperti in quanto richiede di settare alcune opzioni il cui significato potrebbe non essere conosciuto, fortunatamente a tali voci è automaticamente attribuito un valore di default che permette, in un certo senso, di poterle ignorare.

Il processo di setup del software si suddivide in tre fasi. La prima controlla che nel sistema sia già installato il DBMS DB2, il quale rappresenta il perno attorno al quale lavora il programma, e, in caso contrario, ne richiede l'installazione. Successivamente vengono installati i moduli per la realizzazione dei wrapper relazionali e per quelli non relazionali (che permetteranno il collegamento con le rispettive tipologie di sorgenti di dati remote), l'utente ha la possibilità di scegliere se importarne, nel proprio sistema, solo uno o entrambi; ovviamente la seconda scelta permette di poter integrare un maggior numero di sorgenti e quindi risulta essere la più completa e la più ottimale.

A processo terminato si potrebbe incorrere in una situazione di smarrimento causata dal fatto che non si individui nessun nuovo programma installato. Questo è del tutto normale in quanto DB2 Information Integrator non è un software a sé, ma si integra direttamente con DB2 arricchendolo con nuove funzionalità e potenzialità.

## 4.2 Configurazione del sistema federato

Passo successivo all'installazione di DB2 Information Integrator è quello che prevede la registrazione delle sorgenti di dati all'interno di un database. Qui di seguito vengono riportati i passaggi fondamentali di questo procedimento. Ognuno di essi può essere ottenuto sia attraverso l'interfaccia grafica del DB2 Control Center, in modo più o meno banale, sia attraverso l'esecuzione di codice SQL tramite il Command Center. Nella trattazione si darà maggiore importanza a questo secondo approccio, per una più esauriente descrizione del primo si rimanda alla lettura dei vari tutorial presenti sul sito dell'IBM (vedere bibliografia) che chiariscono in modo ottimale come realizzare i vari passi che portano all'integrazione delle varie tipologie di sorgenti.

Per permettere una migliore comprensione degli argomenti esposti, l'esposizione farà riferimento all'esempio qui di seguito riportato.

Supponiamo di dover gestire un tipico negozio on line; la struttura del sistema si basa su di un warehouse che contiene tutti i dati dei clienti attraverso l'utilizzo di DB2 in un tabella denominata *Clienti*. La sede centrale in cui è situato il database principale riceve ordini sottoforma di documenti XML ed è collegata a due data warehouse relativi alle due filiali della società. Essi memorizzano informazioni inerenti ai prodotti e ai fornitori in tabelle chiamate rispettivamente *prodotti* e *fornitori*

e possiedono, inoltre, una terza entità, *prod_forn*, che contiene i collegamenti tra i vari record di queste due tabelle. I due database in questione sono rispettivamente implementati in SQL Server e Microsoft Access.

La figura 11 rappresenta questo scenario.

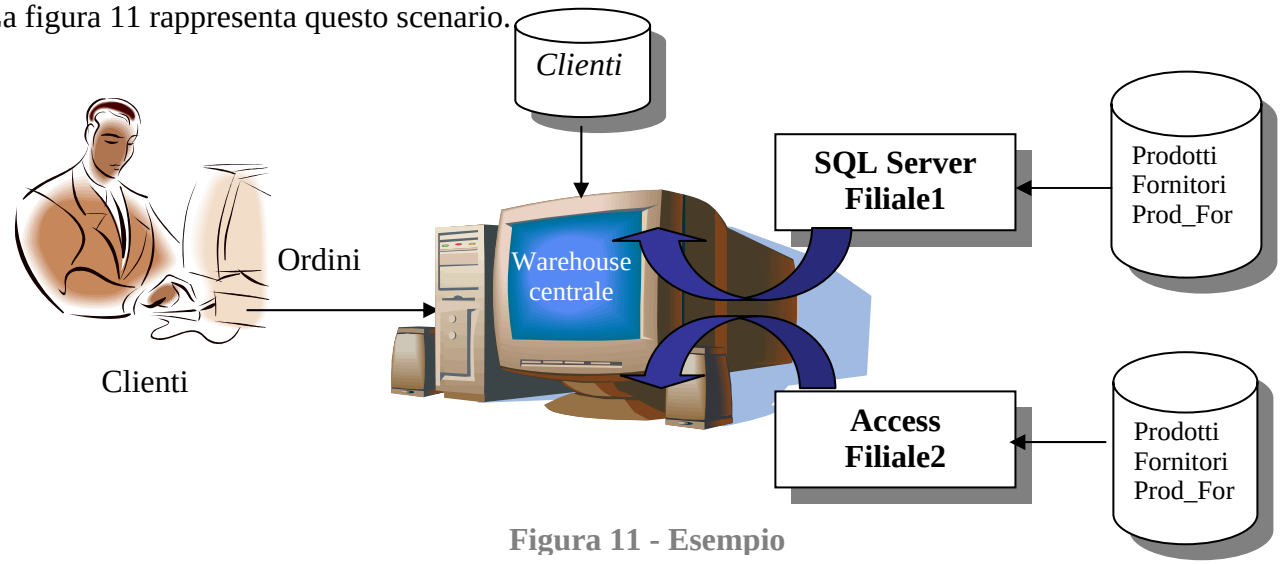



**Figura 11 - Esempio**

## 4.2.1 I Wrapper

Il server federale comunica con le sorgenti di dati attraverso moduli software chiamati wrapper i quali hanno lo scopo di agevolare le seguenti operazioni:

- Registrazione degli oggetti: Un wrapper è a conoscenza di tutte le informazioni necessarie per registrare ogni tipo di sorgente e contiene le caratteristiche di ognuna di esse.
- Comunicazione con le sorgenti di dati: La comunicazione comprende tutte le fasi che vanno dall'instaurazione alla chiusura del processo di connessione con una data sorgente, comprendendo, dunque, anche tutte le operazioni di mantenimento di tale collegamento.
- Operazioni e servizi: Sono previste diverse operazioni in base alle caratteristiche della sorgente con cui il wrapper deve interagire, esse possono essere, ad esempio, l'invio di query per ottenere informazioni, l'aggiornamento dei dati remoti, transazioni, la manipolazione di oggetti ecc.
- Modellazione dei dati: Rappresenta una caratteristica fondamentale di un wrapper ovvero quella di rappresentare i risultati di una query in formato tabulare così come richiesto dal motore federato.

DB2 Information Integrator fornisce un insieme di wrapper per diversi tipi di sorgenti e offre la possibilità di utilizzare il "wrapper development kit", già incluso nel pacchetto, per realizzare una libreria di wrapper personali che permetterà, all'utente, di accedere a sorgenti non supportate da quelli esistenti. È comunque consigliato, per mantenere alto il livello di prestazioni, di utilizzare wrapper implementati nell'API nativo.

Per il nostro esempio i wrapper da creare saranno i seguenti:

```
CREATE WRAPPER "Wrapper_DB2" LIBRARY 'db2drda.dll';

CREATE WRAPPER "Wrapper_SQL" LIBRARY 'db2mssql3.dll';
```

CREATE WRAPPER "Wrapper_Access" LIBRARY 'db2rcodbc.dll';

CREATE WRAPPER "Wrapper_XML" LIBRARY 'db2lsxml.dll';

Va infine ricordato che è sufficiente registrare un unico wrapper per tutte le sorgenti di dati dello stesso tipo.

### 4.2.2 I Server

Un server rappresenta una specifica sorgente di dati alla quale si accede tramite un wrapper. Durante la sua definizione è possibile settare alcune opzioni in modo da gestire le varie interazioni; tali opzioni riguardano attributi che possono essere interni od esterni. Mentre i primi non sono modificabili, i secondi possono essere ritoccati e si riferiscono a informazioni riguardanti la locazione della sorgente di dati (nodo), la sicurezza (userid e password) ed altre caratteristiche che concernano le prestazioni.

Qui di seguito viene riportato il codice SQL per realizzare i server necessari:

CREATE SERVER Server_Clienti TYPE DB2/UDB VERSION '8.1' WRAPPER "Wrapper_DB2" AUTHID "admin" PASSWORD "password" OPTIONS( ADD DBNAME 'e-commerce', PASSWORD 'Y');

CREATE SERVER Server_Filiale1 TYPE MSSQLSERVER VERSION '2000' WRAPPER "Wrapper_SQL" OPTIONS( ADD NODE 'LocalServer', DBNAME 'Filiale1', PASSWORD 'N');

CREATE SERVER Server_Filiale2 TYPE ODBC VERSION '3.0' WRAPPER "Wrapper_Access" OPTIONS( ADD NODE 'Access', PASSWORD 'Y');

I nomi dei nodi si riferiscono ai nomi degli oggetti ODBC relativi alle sorgenti da integrare definiti tramite gli strumenti d’amministrazione di Windows.

CREATE SERVER Server_Ordini WRAPPER "Wrapper_XML";

### 4.2.3 Associazioni Utente

Questa fase permette di stabilire una corrispondenza tra gli username e le password di uno o più utenti di DB2 Information Integrator e i rispettivi username e password delle sorgenti remote. Va sottolineato che non tutte le sorgenti richiedono questo procedimento, ne sono un esempio i file XML per i quali tale procedura non è prevista.

Visto che il mapping risulta essere simile per i vari casi analizzati si riporta solo quello inerente alla tabella *Clienti*.

Le associazioni create riguardano due utenti: User1 e User2.

CREATE USER MAPPING FOR "User1" SERVER "Server_Clienti" OPTIONS ( ADD REMOTE_AUTHID 'Admin', ADD  REMOTE_PASSWORD 'password');

CREATE USER MAPPING FOR "User2" SERVER "Server_Clienti" OPTIONS ( ADD REMOTE_AUTHID 'Admin', ADD  REMOTE_PASSWORD 'password')

### 4.2.4 Sessione pass-through

Una sessione pass-through permette di sottoporre codice SQL direttamente al server remoto. Dopo la definizione del mapping è possibile verificare il corretto funzionamento della connessione con la sorgente dei dati, sempre che questa supporti il pass-through. Effettuare tale controllo in questo momento, che comunque non è necessariamente richiesto e quindi può anche essere evitato, aiuta ad isolare eventuali problemi di configurazione prima della definizione dei “nickname”.

### 4.2.5 I Nickname

Un nickname rappresenta l’alias di una tabella o di una vista appartenente ad una sorgente di dati. Le informazioni inerenti gli oggetti remoti con i quali essi sono in relazione vengono memorizzate nelle tabelle di sistema locali di DB2; in questo modo il DBMS ha la possibilità di conoscere dati fondamentali per il processo di ottimizzazione delle query che permette di ottenere migliori prestazioni, i principali sono: i tipi di dati presenti nelle tabelle remote, la velocità del processore e quella di I/O, il numero di record interessati e la larghezza di banda della rete.

È necessario definire un numero di nickname pari alla quantità di oggetti da importare, i nomi locali corrispondenti saranno quelli che verranno effettivamente utilizzati nelle query come se gli oggetti remoti risiedessero localmente in DB2.

Per l’esempio analizzato sarà necessario definire i seguenti nickname:

```
CREATE NICKNAME SCHEMA.CLIENTE FOR Server_Clienti.SCHEMA.CLIENTI;

CREATE NICKNAME SCHEMA.FORNITORI_F1 FOR Server_Filiale1.Fornitori;
CREATE NICKNAME SCHEMA.PRODOTTO_F1 FOR Server_Filiale1.Prodotti;
CREATE NICKNAME SCHEMA.PROD_FORN_F1 FOR Server_Filiale1.Prod_Forn;

CREATE NICKNAME SCHEMA.FORNITORI_F2 FOR Server_Filiale2."Fornitori";
CREATE NICKNAME SCHEMA.PRODOTTO_F2 FOR Server_Filiale2."Prodotto";
CREATE NICKNAME SCHEMA.PROD_FORN_F2 FOR Server_Filiale2."Prod_Forn";
```

SCHEMA rappresenta il nome dello schema in cui si vogliono importare i nickname.

```
CREATE NICKNAME xml.ordini
        (id             char (10)       OPTIONS (XPATH ‘@id ’),
        data            date            OPTIONS (XPATH ‘date/text()’),
        idCliente       char (10)       OPTIONS (XPATH ‘cid ’),
        PrezzoTotale    money           OPTIONS (XPATH ‘./amount/text()’),
        oid             varchar (16)    OPTIONS (PRIMARY_KEY ‘YES ’))
FOR SERVER Serve_XML
OPTIONS (FILE_PATH ‘/Documenti/tesi/ordini.xml ’, XPATH ’//ordini ’);

CREATE NICKNAME xml.ordini_prodotti
        (oid            varchar (16)    OPTIONS (FOREIGN_KEY ‘ORDINI ’),
        idProdotto      char(10)        OPTIONS (XPATH ‘./item_id/text()’),
        Quantità        integer         OPTIONS (XPATH ‘./quantity/text()’))
```

FOR SERVER Server_XML
OPTIONS (XPATH ‘.//prodotti ’);

### 4.2.5.1 Personalizzazione dei nickname

Eseguendo la semplici istruzioni SQL così come mostrato precedentemente o ricorrendo all’interfaccia grafica, il sistema definisce automaticamente la struttura dei nickname così come è si presenta nella tabella o nella vista remota. Questo, in alcuni casi, potrebbe rappresentare un limite in quanto per necessità di integrazione potrebbe essere necessario che determinati campi abbiano un nome specifico o un tipo diverso da quello col quale sono stati originariamente creati. Esiste dunque uno strumento che consente di effettuare queste modifiche e che viene qui di seguito mostrato.

Il codice seguente è inerente ad una personalizzazione del nickname *Clienti* per il quale sono stati modificati sia il tipo di dato (per la colonna *nome*), sia il nome di una colonna (*telefono*):

ALTER NICKNAME "SCHEMA"."CLIENTI" ALTER COLUMN NOME LOCAL TYPE VARCHAR (100) ALTER COLUMN TELEFONO LOCAL NAME TEL ;

### 4.2.6 Funzioni delle sorgenti

È possibile, infine, realizzare un mapping tra le funzioni tipiche di DB2 e quelle delle sorgenti remote purché esse siano relazionali. In alcuni casi è strettamente necessario ricorrere a determinate funzioni che potrebbero non essere fornite dal DBMS dell’IBM ma che sono previste dalla sorgente remota. In questa circostanza è possibile definire una “function template” che può essere considerata come un’estensione di una funzione definita dall’utente. In figura 12 viene mostrato il processo logico che sta alla base di questa possibilità e che permette di chiarire meglio il problema.

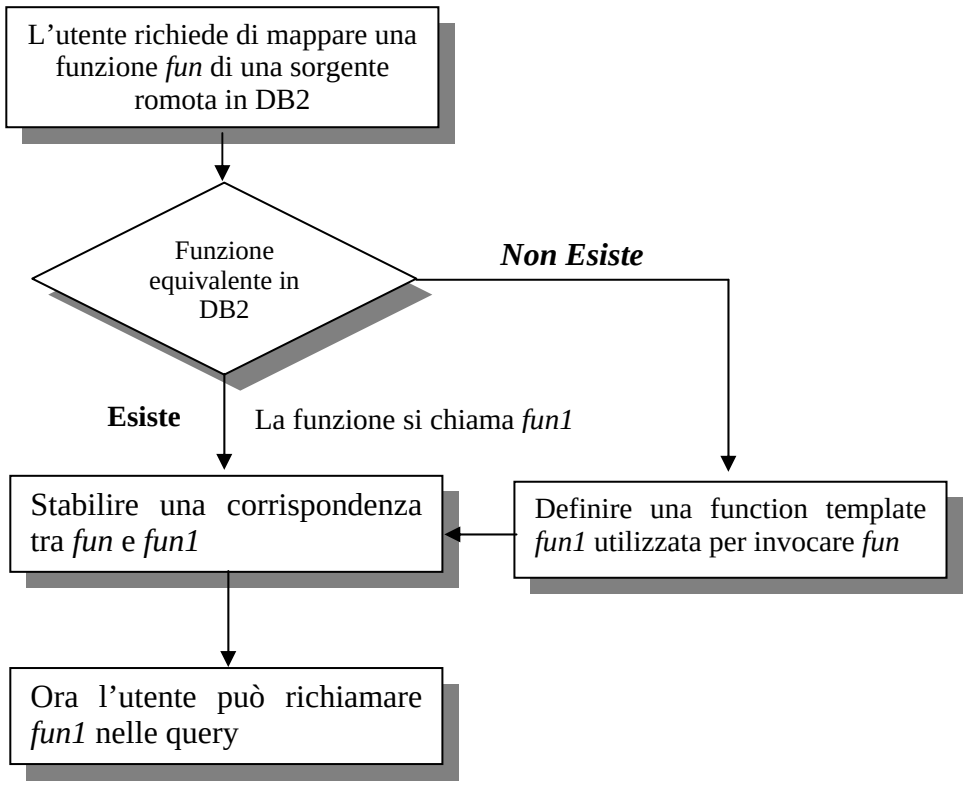



**Figura 12 – Function template**

Le sorgenti non relazionali, generalmente, hanno un insieme di funzioni prefissato che il wrapper riesce a riconoscere e questo rende insolita la necessità di ricorrere a questo procedimento.

### 4.2.7 DB2 System Catalog

Tutte le varie fasi finora descritte memorizzano le informazioni nel DB2 system catalog il quale può essere visto come una tabella locale tramite la quale è possibile osservare tutti i dati relativi agli oggetti remoti oltre che alle varie relazioni esistenti tra i wrapper, i server e i nickname. La figura 13 mostra lo schema a blocchi dell'intera struttura federata e rappresenta, inoltre, un sunto visivo di quanto finora descritto.

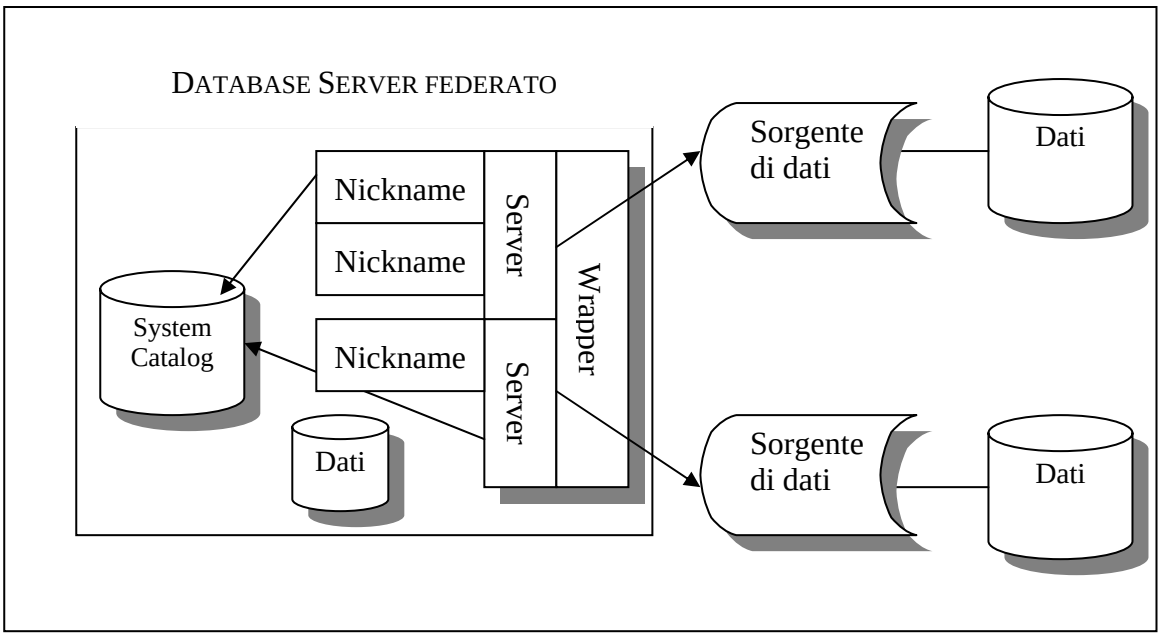



*Figura 13 - I componenti federati DB2 di base*

## 4.3 Test dell'ambiente creato

Per testare l'ambiente creato si è sottoposta l'intera struttura ad una serie di query atte a verificare l'effettiva integrazione delle varie sorgenti. Come riscontro di quanto appena dichiarato viene riportata una di queste interrogazioni con il relativo risultato.

```
select  c.nome as nome_cliente, p.nome as nome_prodotto
from    customer c
        join acq as a on c.id=a.idc
        join product as p on a.idp=p.id
        join mix1 as pf on p.id=pf.idp
        join forn as f on pf.idf=f.id
where   f.nome='Franzini SPA' or f.nome='Litograf 5 SNC'
```

Dove global_prodotti, global_fornitori e global_prod_for sono viste realizzate per fondere insieme le informazioni provenienti dalle due filiali e così definite:

```
create view global_prodotti as
select *
from prodotti_f1
union
select *
from prodotti_f1
```

```
create view global_fornitori as
select *
from fornitori_f1
union
select * from
fornitori_f2
```

```
create view global_prod_for as
select *
from prod_for_f1
union
select *
from prod_forn_f2
```

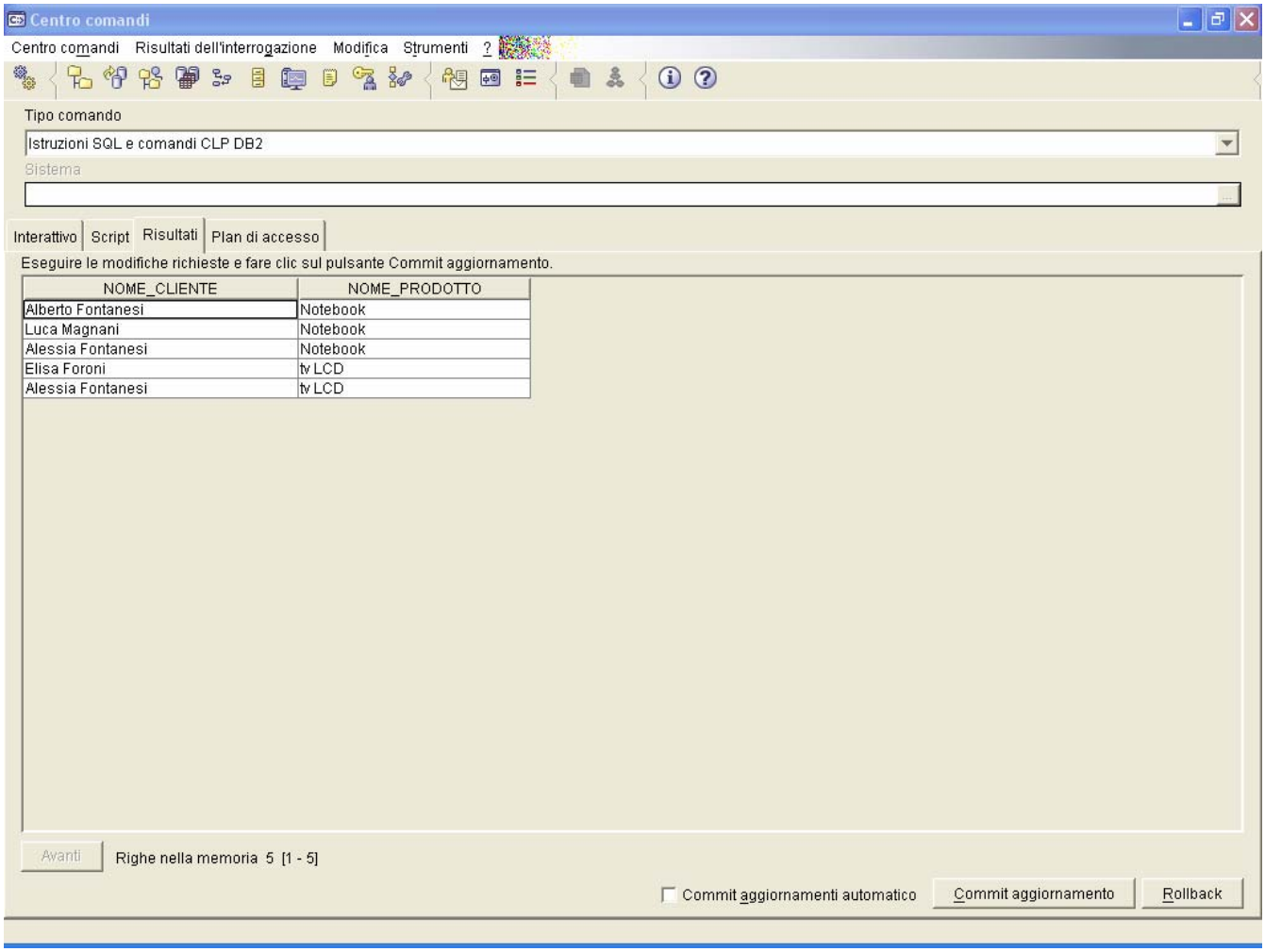



*Figura 14 – Risultati Select di prova*

# MOMIS

MOMIS (Mediator envirOnment for Multiple Information Sources) è un framework per l'estrazione e l'integrazione di informazioni appartenenti a sorgenti dati strutturate e semistrutturate. Per compiere l'estrazione viene introdotto un linguaggio object-oriented con una semantica basata su di una Description Logics chiamata $ODL_I^3$ derivata dallo standard ODMG. L'integrazione delle informazioni viene compiuta in modo semi-automatico, impiegando la conoscenza presente in un Common Thesaurus (definito utilizzando il framework), le descrizioni $ODL_I^3$ degli schemi sorgenti, tecniche di clustering e, appunto, di Description Logics. Il processo di integrazione definisce un vista virtuale integrata degli schemi sottostanti (chiamata Global Schema) nella quale sono specificate regole di mapping e vincoli di integrità per la gestione delle eterogeneità. Il sistema MOMIS, basato sull'architettura wrapper/mediator, fornisce le modalità e tool aperti per il data management in Internet-based information systems utilizzando un'interfaccia compatibile CORBA-2. MOMIS è stato sviluppato come collaborazione tra il DBGroup dell'Università degli Studi di Modena e Reggio Emilia e l'Università di Milano e Brescia nell'ambito del progetto nazionale INTERDATA nel periodo che va dal 1997 al 1999 e successivamente in quello del progetto D2I (2000-2002), sotto la direzione della Professoressa S. Bergamaschi. Ora l'attività di analisi continua all'interno del progetto di ricerca europeo SEWASIE col quale si prevede una collaborazione fino al 2005.

## 5.1 L'architettura MOMIS

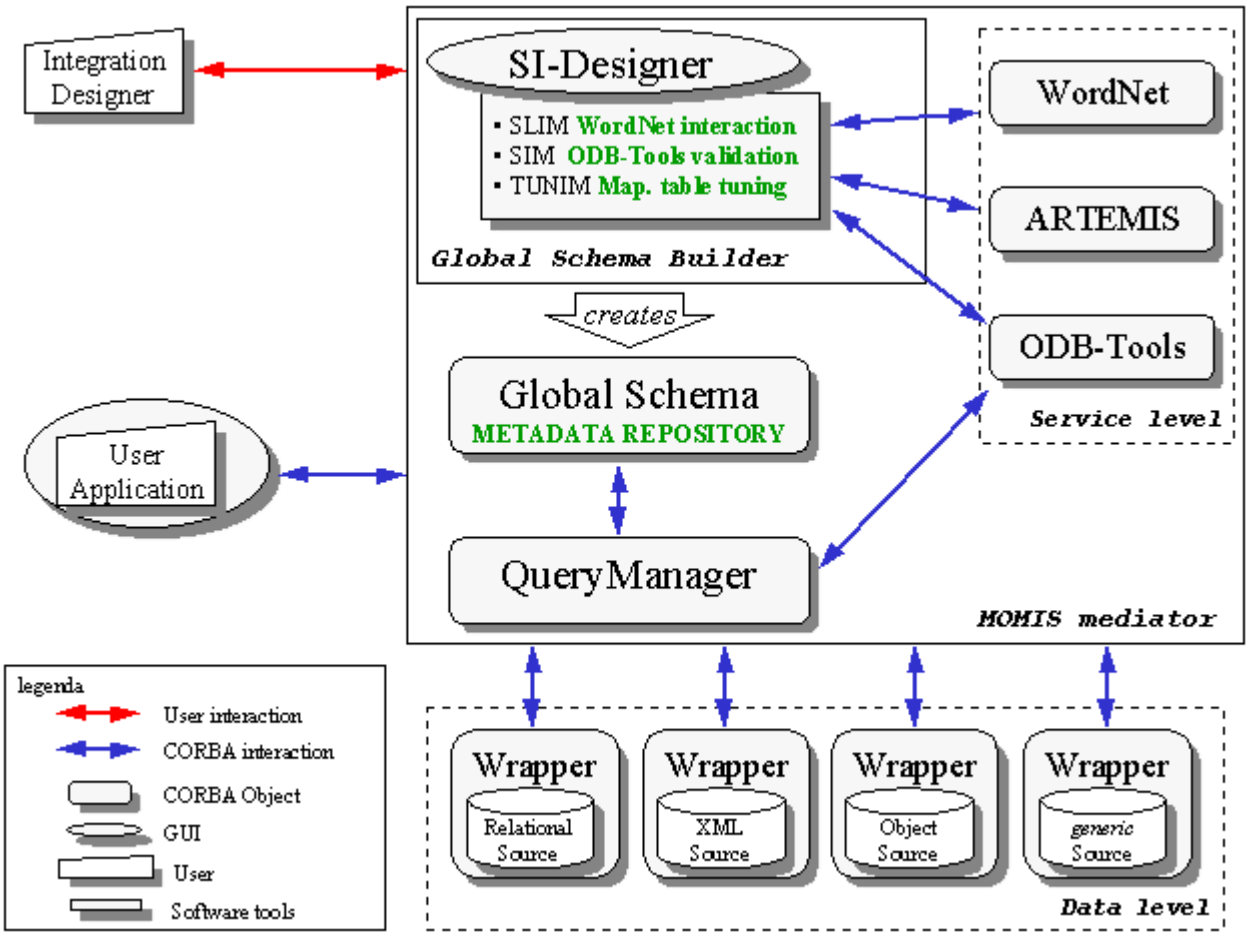



**Figura 15 – Struttura a blocchi del sistema MOMIS**

Il tool è composto da:

- Server
  - Global schema
  - Query Manager
- Interfaccia grafica per gli utenti (GUI)
  - SI-Designer
    - SIM Schemata Integrator Module
    - SLIM Source Lessical Integration Module, (interaction with the Wordnet Ontology)
    - ARTEMIS Affinity calculus and Clustering
    - TUNIM Mapping-table Tuning
- Wrapper
  - MOMIS Wrapper XML
  - MOMIS Wrapper JDBC
- Tool module
  - CORBA WordNet
  - CORBA ODB-Tool
  - CORBA Artemis

L'intera struttura poggia su di un modello di dati comune ($ODM_I^3$) il quale è definito conformemente al linguaggio $ODL_I^3$ per la descrizione degli schemi delle sorgenti. Entrambi sono stati definiti, all'interno del sistema, come sottoinsieme di quelle corrispondenti in ODMG seguendo la proposta di un linguaggio di mediazione standard sviluppato da gruppo di lavoro $I^3$/POB.

### 5.1.1 Global Schema

Il Global Schema è l'oggetto CORBA che permette l'accesso alle informazioni di uno schema integrato di MOMIS. Attraverso l'interfaccia grafica SI-Designer, il progettista ha la facoltà di integrare le varie sorgenti; tutte le informazioni create con essa vengono poi salvate in un oggetto CORBA Global Schema object che permetterà al Query Manager di effettuare interrogazioni sullo schema integrato.

### 5.1.2 Query Manager

Il modulo Query Manager esegue il query processing e l'ottimizzazione generando le query $OQL_I^3$ per i wrapper a partire da quelle formulate dall'utente sullo schema globale. Utilizzando le tecniche di Description Logics, il componente genera in modo automatico la traduzione delle interrogazioni globali nelle sotto-query utili a ciascuna sorgente locale coinvolta, quindi invia tali query alle sorgenti, attende le risposte calcolandone una, singola ed unificata, da presentare all'utente.

### 5.1.3 SI-Designer

SI-Designer è la GUI relativa alla parte di creazione del Global Schema del sistema MOMIS. Essa guida l'utente attraverso le varie fasi dell'integrazione, dall'acquisizione delle sorgenti fino alla messa a punto delle tabelle di mapping. Si tratta di un contenitore modulare di altri strumenti che permettono la raccolta di informazioni per l'integrazione e la dichiarazione delle relazioni tra gli oggetti degli schemi.
SI-Designer è composto da quattro moduli

- SIM (Source Integrator Module): estrae le relazioni intra-schema di una sorgente relazionale, ad oggetti o semi-strutturata. Inoltre effettua la "validazione semantica" delle relazioni e ne inferisce delle nuove sfruttando ODB-Tool.

- SLIM (Sources Lexical Integrator Module): estrae le relazioni inter-schema tra nomi di classi e di attributi di differenti sorgenti, sfruttando il sistema lessicale WordNet.
- ARTEMIS (Analysis and Reconciliation Tool Environment for Multiple Information Sources): tool che implementa tecniche di clustering basate sull'affinità delle classi.
- TUNIM (Tuning of the mapping Table): modulo che gestisce la fase di creazione dello schema globale.

La GUI di SI-Designer è una sequenza di finestre, ognuna delle quali relativa ad una fase del processo di integrazione. Il tool è stato progettato ed implementato in Java e un enorme sforzo è stato fatto per sviluppare un'architettura modulare, in modo da rendere molto semplice l'aggiunta di nuove fasi nel processo di integrazione. Esso viene reso possibile grazie ad un approccio semantico il quale ricorre a tecniche basate sulle logiche di descrizione intelligenti OLCD le quali, assieme ad un modello esteso ODM-ODMG rappresentano le informazioni estratte ed integrate $ODM_I^3$. Lo strumento assiste il programmatore nella creazione di una vista integrata di tutte le sorgenti necessarie ( chiamata Global Virtual View); essa è espressa con lo standard XML il quale permette di andare incontro alle necessità di scambiare dati tra i processi aziendali e le opportune applicazioni a prescindere dai problemi legati alle singole sorgenti o alle destinazioni.

#### *5.1.3.1 SIM*

Questo modulo è stato uno dei primi componenti MOMIS sviluppati. Da poco reimplementato, oggi SIM si presenta come strumento di inserimento di relazioni all'interno del Common Thesaurus. Esso è in grado di estrarre relazioni tramite la conoscenza della struttura degli schemi, validare quelle inserite dal progettista e di calcolare (grazie al motore deduttivo ODB-Tool) tutte le possibili relazioni implicate partendo da quelle esplicite contenute nel Common Thesaurus.

#### *5.1.3.2 WordNet*

WordNet è un sistema lessicale refenziale on-line il cui progetto si ispira alle attuali teorie psicolinguistiche della memoria lessicale umana. I nomi, gli aggettivi e gli avverbi di lingua inglese sono organizzati in insiemi di sinonimi, ciascuno dei quali rappresenta un concetto lessicale che sottintende. WordNet è stato sviluppato presso i Cognitive Science Laboratory della Princeton University sotto la direzione del Professor George A. Miller. Nel framework MOMIS, WordNet è utilizzato attraverso l'interfaccia CORBA-2 per l'estrazione delle relazioni intensionali inter-schema tra i concetti.

#### *5.1.3.3 ARTEMIS*

ARTEMIS è un tool, sviluppato presso l'università di Milano e Brescia, basato sulle tecniche di affinità e di clustering. Il modulo effettua, in primo luogo, l'analisi semantica, la quale ha come obiettivo l'identificazione degli elementi che sono tra loro in relazione, in base al proprio significato, nei diversi schemi. Il concetto di affinità viene introdotto per valutare, appunto, il livello di relazione semantica tra gli elementi di uno schema e si appoggia all'utilizzo di coefficienti di affinità, calcolati tra coppie di elementi, per esprimere la loro similarità nel rappresentare la stessa informazione in schemi differenti. Il passo successivo è rappresentato da un processo di clustering gerarchico il cui fine è quello di identificare e raggruppare tutti gli elementi che presentano le suddette affinità.

#### *5.1.3.4 TUNIM*

Questo modulo gestisce l'ultima fase del procedimento di integrazione per la creazione dello schema globale. Partendo dalle relazioni del Common Thesaurus, per ciascuno dei cluster individuati da ARTEMIS viene creata una classe globale. Ognuna di queste è caratterizzata da un insieme di attributi globali e da una tabella di mapping: i primi ne definiscono la struttura mentre la seconda indica quali informazioni locali sono mappate in ogni attributo globale.

### 5.1.4 Wrapper

Come già detto i wrapper sono strumenti che permettono il collegamento nonché le interazioni fra il sistema locale e le sorgenti di dati. Il loro compito è quello di tradurre le descrizioni dei metadati remoti nella rappresentazione comune $ODL_I^3$ e di riformulare le query globali, espresse nel linguaggio d'interrogazione $OQL_I^3$ (che rappresenta un sottoinsieme di OQL-ODMG), in quello originario delle sorgenti permettendo così di ottenere le informazioni richieste.

## 5.2 Classi globali e tabelle di mapping

Una volta ottenuto l'output del processo di clustering è possibile, partendo da esso, definire, per ogni cluster, una classe globale la quale rappresenta una vista unificata di tutte le classi del cluster. Per ognuna di esse viene fornito, dal sistema, un insieme di attributi globali che corrispondono, attraverso le regole di mapping, ai singoli attributi locali. In termini molto semplicistici si può pensare che tali attributi globali vengano ottenuti in due step durante i quali vi è un'eliminazione delle ridondanze semi-automatica basata sulle relazioni contenute nel Common Thesaurus. Tali passaggi sono:

- l'unione degli attributi di tutte le classi appartenenti al cluster
- l'aggregazione degli attributi "simili"

Tutte le relazioni che si vanno così a formare entrano a far parte della tabella di mapping; in essa le colonne rappresentano l'insieme delle classi locali che appartengono al cluster, le righe, al contrario, contengono gli attributi globali. Un elemento $MT[L][ag]$ rappresenta come l'attributo globale *ag* sia mappato nella classe locale *L*, e può assumere i seguenti valori:

- $MT[L][ag] = al$: l'attributo globale *ag* corrisponde a quello locale *al*
- $MT[L][ag] = al_1$ and $al_2$ and...and...$al_n$: il valore di *ag* è dato dalla concatenazione dei valori assunti da un insieme di attributi $al_i$ appartenenti alla stessa classe locale *L*
- $MT[L][ag]$ = case of *al* $const_1:al_1,...,const_n:al_n$: come prima *ag* può assumere un valore all'interno di un insieme di attributi $al_i$ appartenenti alla stessa classe locale *L*, ma in questo caso il valore scelto dipende da un terzo attributo, *al*, della stessa classe, il quale funge da selettore
- $MT[L][ag]$ = const: *ag* non corrisponde a nessun attributo locale bensì ad un valore costante assegnatogli dal progettista
- $MT[L][ag]$ = null: ad *ag* non viene assegnato alcun valore e questo, ovviamente, implica anche la sua completa indipendenza gli attributi locali di *L*

## 5.3 Processo d'integrazione

Il processo d'integrazione consiste in un insieme di step implementati in moduli separati che vedono la loro unificazione nel framework Si-Designer.
Qui di seguito vengono riportate queste varie fasi:

- *Acquisizione delle sorgenti*: stadio in cui l'utente stabilisce le sorgenti da importare, un wrapper permette la traduzione del modello di descrizione della sorgente in quello $ODL_I^3$.
- *Definizione delle relazioni intensionali*: fase in cui vengono aggiunte nuove relazioni, attraverso l'interazione con il modulo SIM, il sistema ODB-Tool e WordNet, al Common Thesaurus.
- *Definizione delle relazioni estensionali*: Le relazioni estensionali sono definite dall'interazione con il programmatore e vengono sfruttate per rilevare eventuali classi sovrapposte estensionalmente.
- *Clustering*: in questa fase, basata sulla conoscenza presente nel Common Thesaurus, vengono create le classi globali tramite l'utilizzo di ARTEMIS.

- *Mapping Table tuning*: grazie a questa funzionalità l'utente ha la possibilità di modificare la Global Virtual View proposta automaticamente dal sistema per ogni classe globale generata al passo precedente.

La fase finale di questo processo d'integrazione permette l'esportazione della Global Virtual View in un DTD XML preoccupandosi di aggiungere i tag necessari per rappresentare le relazioni tra le tabelle di mapping. L'utilizzo di XML nella definizione della vista globale garantisce la compatibilità di MOMIS con altri sistemi d'integrazione dell'informazione nello scambio di file di dati nel suddetto formato; in più la possibilità di tradurre il Common Thesaurus in un file XML permette al tool di fornire un'ontologia condivisa che può essere utilizzata da linguaggi con ontologie semantiche differenti.

# Sperimentazione di MOMIS

Questo capitolo è dedicato alla descrizione del processo d'integrazione in ambiente MOMIS. L'analisi è stata effettuata riproducendo scenari già realizzati dal gruppo di lavoro dell'Università degli Studi di Modena e Reggio Emilia che consistono nell'integrazione di informazioni inerenti ad aziende del territorio locale.

## 6.1 Installazione e preparazione dell'ambiente di lavoro

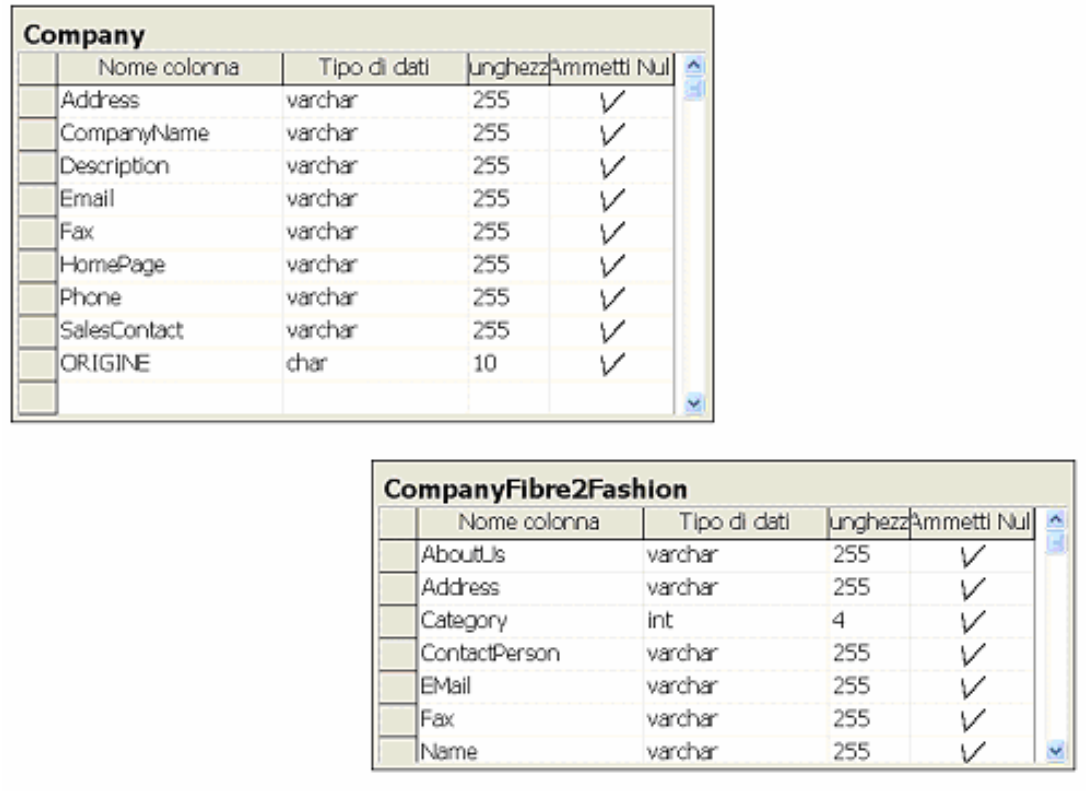
Sotto questo punto di vista il software presenta forse il suo maggior difetto in quanto necessita di due DBMS, MySQL ed SQL Server, per la realizzazione di due funzioni differenti. Il DBMS della Microsoft è utilizzato per permettere la realizzazione delle query; è necessario creare, al suo interno, un database vuoto di nome *momis* con user *momis* e password *momis* per consentire, appunto, questa fase fondamentale. La richiesta di MySQL invece riguarda lo stadio di annotazione ovvero quello di attribuzione dei significati ai vari attributi. È necessario innanzitutto aggiungere, tramite le seguenti linee di codice, un record alla tabella *user* del database di sistema *mysql* in modo da settare i privilegi d'accesso e, successivamente, definire un altro db *momiswn* da popolare tramite il comando mysql con un file dump che non viene riportato a causa della sua eccessiva dimensione.

```
INSERT INTO user ( Host, User ) VALUES ( 'localhost', 'momiswn' );
  #
  UPDATE user SET
    Select_priv     = 'Y',
    Insert_priv     = 'Y',
    Update_priv     = 'Y',
    Delete_priv     = 'Y',
    Create_priv     = 'N',
    Drop_priv       = 'N',
    Reload_priv     = 'N',
    Shutdown_priv   = 'N',
    Process_priv    = 'N',
    File_priv       = 'N',
    Grant_priv      = 'N',
    References_priv = 'N',
    Index_priv      = 'N',
    Alter_priv      = 'N'
  WHERE user='momiswn';
  #
  UPDATE user SET Password=PASSWORD('momis') WHERE user='momiswn';
  #
  INSERT INTO db (
     Host,
```

```
    Db,
    User,
    Select_priv,
    Insert_priv,
    Update_priv,
    Delete_priv,
    Create_priv,
    Drop_priv,
    Grant_priv,
    References_priv,
    Index_priv,
    Alter_priv
  ) VALUES ('localhost', 'momiswn', 'momiswn','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y' );
  #
  #
  FLUSH PRIVILEGES;
```

## 6.2 Realizzazione del sistema integrato

Una volta realizzati i database necessari è possibile passare alle varie fasi d'integrazione dei dati attraverso un'interfaccia grafica semplice ed intuitiva. Durante il processo di sperimentazione si è provato ad integrare tra loro varie sorgenti; come esempio si riportano i passi necessari per realizzare una GVV ottenuta da tre database di SQL Server di cui si riporta la struttura.



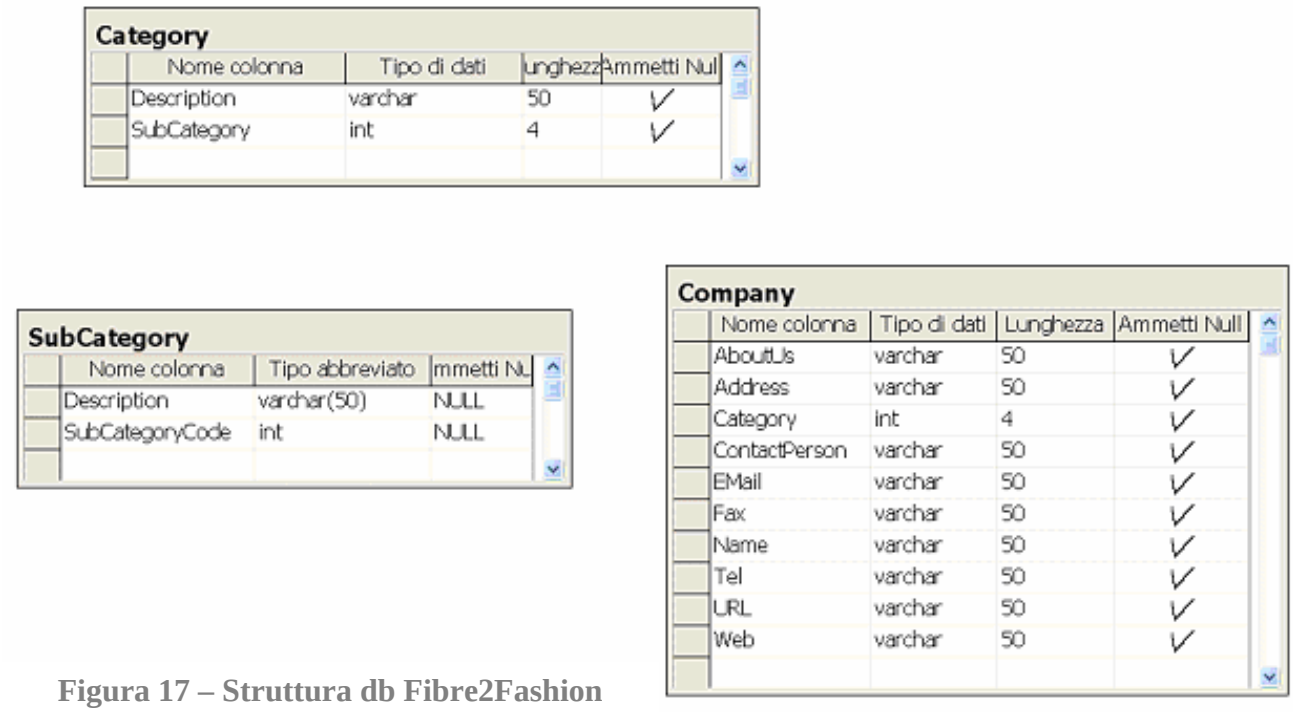
**Company**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| Address | varchar | 255 | ✓ |
| CompanyName | varchar | 255 | ✓ |
| Description | varchar | 255 | ✓ |
| Email | varchar | 255 | ✓ |
| Fax | varchar | 255 | ✓ |
| HomePage | varchar | 255 | ✓ |
| Phone | varchar | 255 | ✓ |
| SalesContact | varchar | 255 | ✓ |
| ORIGINE | char | 10 | ✓ |

**CompanyFibre2Fashion**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| AboutUs | varchar | 255 | ✓ |
| Address | varchar | 255 | ✓ |
| Category | int | 4 | ✓ |
| ContactPerson | varchar | 255 | ✓ |
| EMail | varchar | 255 | ✓ |
| Fax | varchar | 255 | ✓ |
| Name | varchar | 255 | ✓ |

**Figura 16 – Struttura db UsaWear**



**Category**

| | Nome colonna | Tipo di dati | unghezz | Ammetti Null |
|---|---|---|---|---|
| | Description | varchar | 50 | ✓ |
| | SubCategory | int | 4 | ✓ |

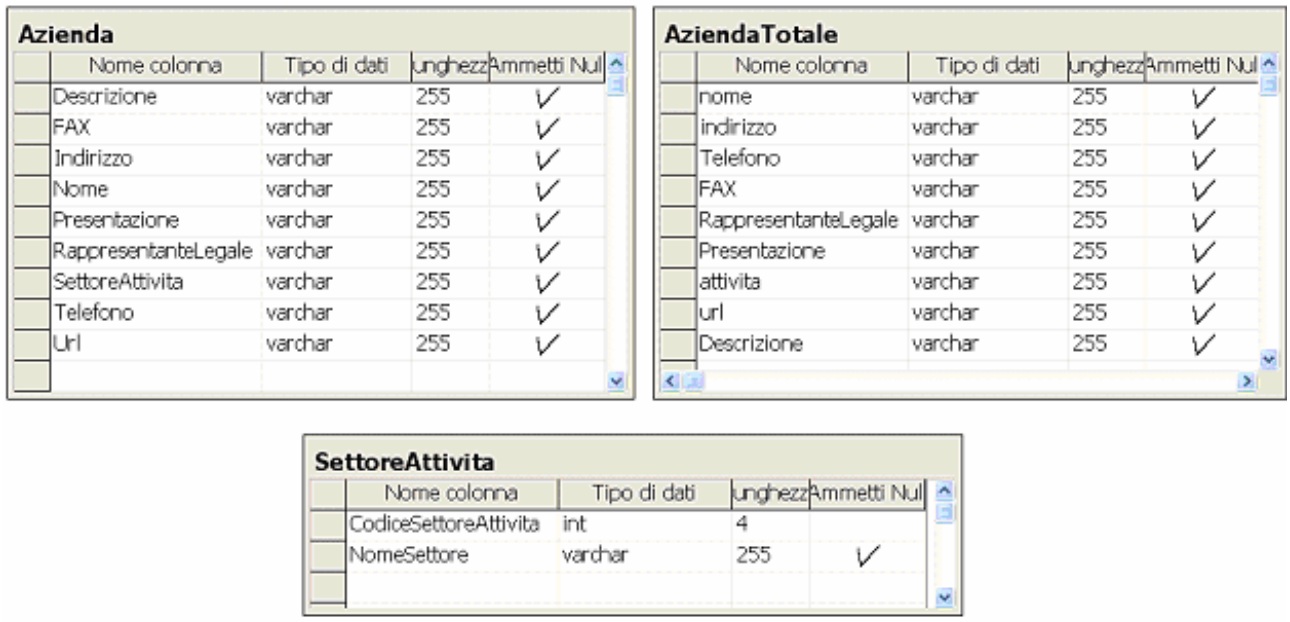
**SubCategory**

| | Nome colonna | Tipo abbreviato | mmetti Nu |
|---|---|---|---|
| | Description | varchar(50) | NULL |
| | SubCategoryCode | int | NULL |

**Company**

| | Nome colonna | Tipo di dati | Lunghezza | Ammetti Null |
|---|---|---|---|---|
| | AboutUs | varchar | 50 | ✓ |
| | Address | varchar | 50 | ✓ |
| | Category | int | 4 | ✓ |
| | ContactPerson | varchar | 50 | ✓ |
| | EMail | varchar | 50 | ✓ |
| | Fax | varchar | 50 | ✓ |
| | Name | varchar | 50 | ✓ |
| | Tel | varchar | 50 | ✓ |
| | URL | varchar | 50 | ✓ |
| | Web | varchar | 50 | ✓ |

**Figura 17 – Struttura db Fibre2Fashion**



**Azienda**

| | Nome colonna | Tipo di dati | unghezz | Ammetti Null |
|---|---|---|---|---|
| | Descrizione | varchar | 255 | ✓ |
| | FAX | varchar | 255 | ✓ |
| | Indirizzo | varchar | 255 | ✓ |
| | Nome | varchar | 255 | ✓ |
| | Presentazione | varchar | 255 | ✓ |
| | RappresentanteLegale | varchar | 255 | ✓ |
| | SettoreAttivita | varchar | 255 | ✓ |
| | Telefono | varchar | 255 | ✓ |
| | Url | varchar | 255 | ✓ |

**AziendaTotale**

| | Nome colonna | Tipo di dati | unghezz | Ammetti Null |
|---|---|---|---|---|
| | nome | varchar | 255 | ✓ |
| | indirizzo | varchar | 255 | ✓ |
| | Telefono | varchar | 255 | ✓ |
| | FAX | varchar | 255 | ✓ |
| | RappresentanteLegale | varchar | 255 | ✓ |
| | Presentazione | varchar | 255 | ✓ |
| | attivita | varchar | 255 | ✓ |
| | url | varchar | 255 | ✓ |
| | Descrizione | varchar | 255 | ✓ |

**SettoreAttivita**

| | Nome colonna | Tipo di dati | unghezz | Ammetti Null |
|---|---|---|---|---|
| | CodiceSettoreAttivita | int | 4 | |
| | NomeSettore | varchar | 255 | ✓ |

**Figura 18 – Struttura db Tessilmoda**

## *6.2.1 I Wrapper*

Il primo step da effettuare è quello dell'importazione dei dati; essa viene realizzata attraverso la scelta del wrapper corrispondente alla sorgente in esame e alla definizione di alcune proprietà necessarie per realizzare, appunto, tale collegamento. In figura 19 viene riportata la schermata inerente all'acquisizione del database fibre2fashion.

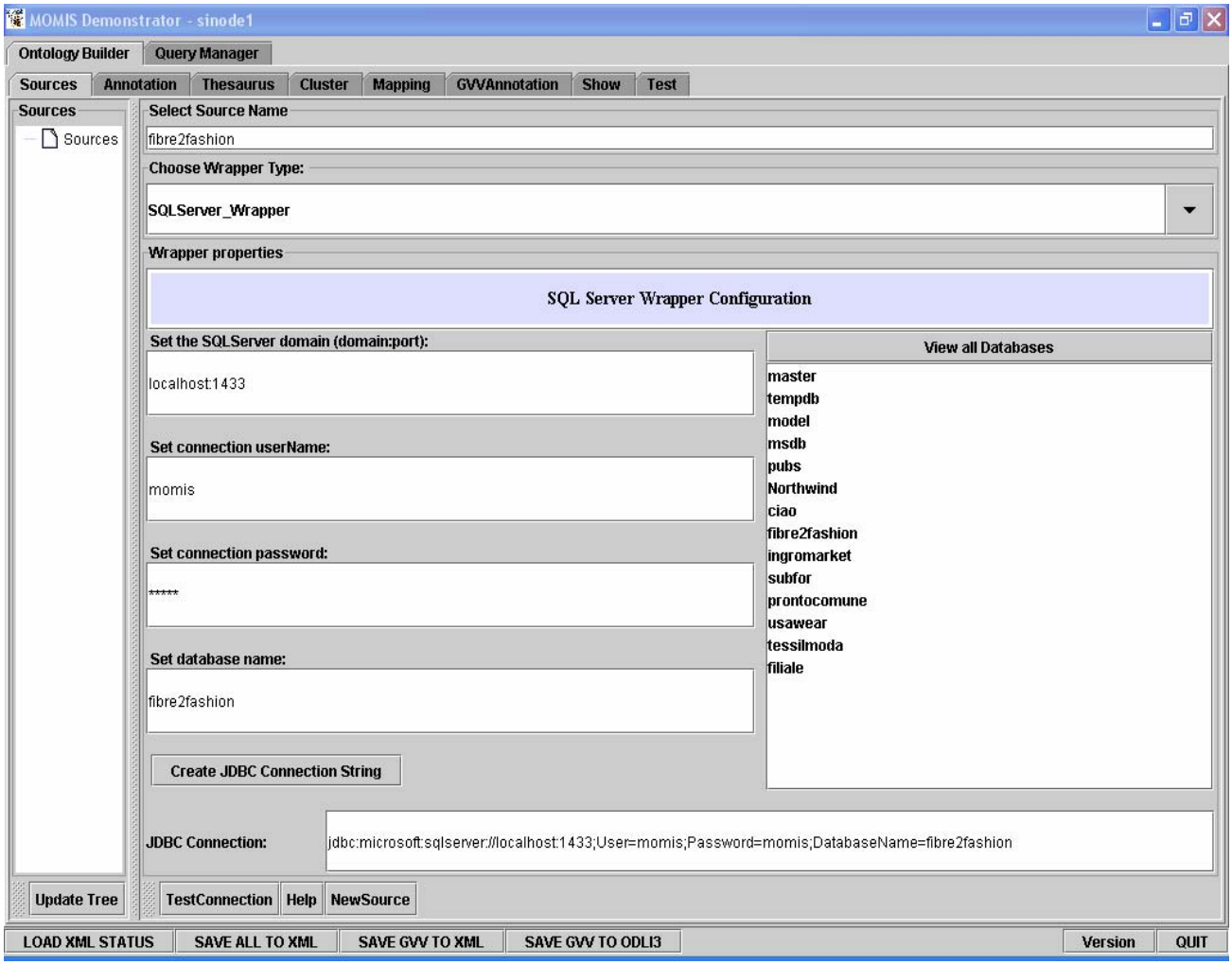



**Figura 19 – Acquisizione sorgenti in MOMIS**

## *6.2.2 Fase di Annotazione*

In questa fase è possibile attribuire, ad ogni oggetto (tabella o attributo), il significato che esso assume all'interno dell'ambiente integrato; ad esempio per indirizzo si dovrà indicare che rappresenta il luogo in cui ha sede l'azienda. Come già detto, l'utente, in questa fase, è supportato dal Common Thesaurus che contiene una serie di definizioni per ogni parola presente al suo interno. Il fatto che esso faccia riferimento a vocaboli in lingua inglese rende necessario assegnare ad ogni nome il corrispettivo termine anglosassone tramite il comando Word Form presente nel menù a tendina che appare cliccando col tasto destro del mouse sull'oggetto interessato.

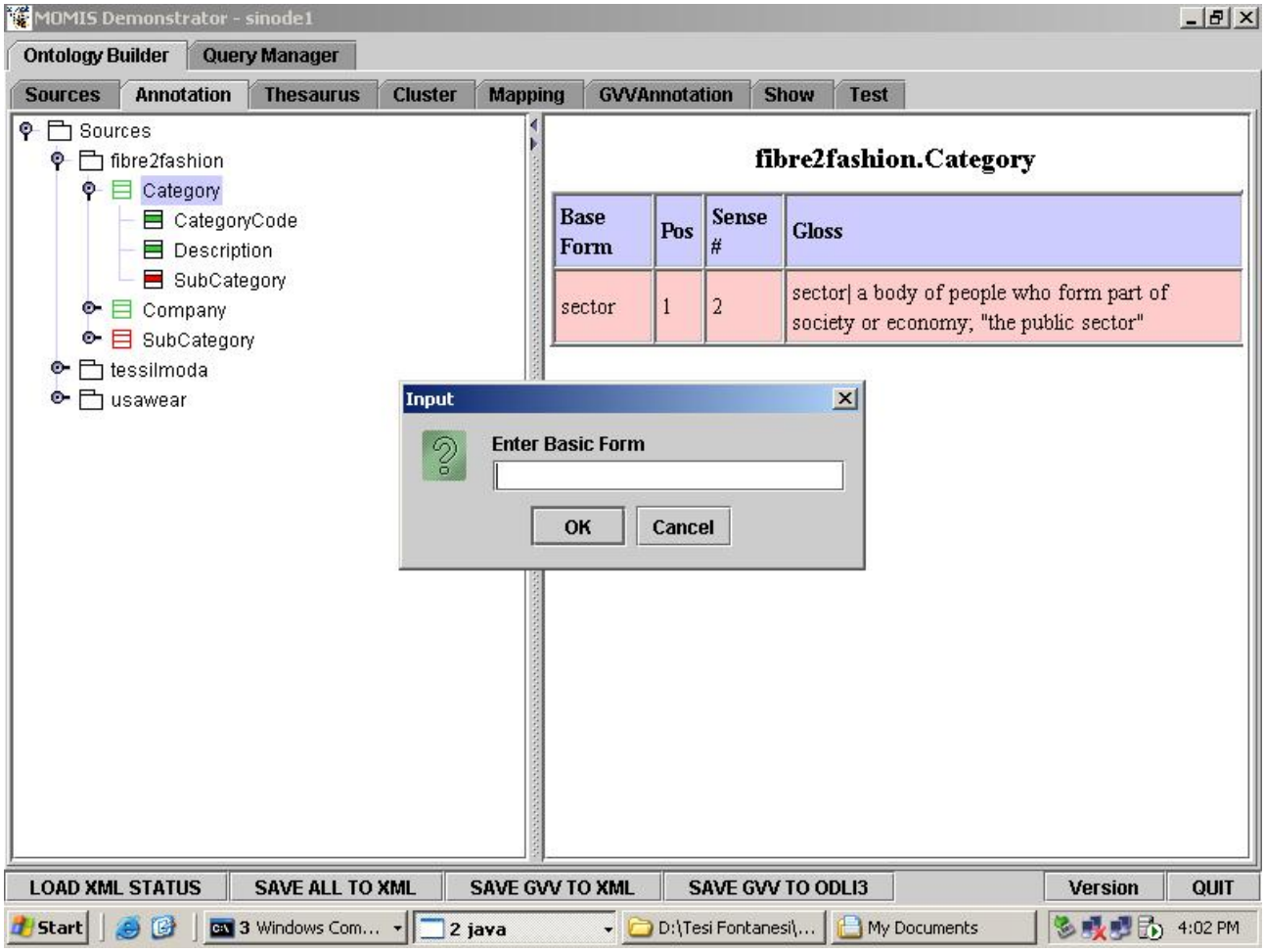



**Figura 20 – Assegnazione dei termini inglesi**

Successivamente si deve assegnare il significato all'oggetto compiendo una scelta tra quelli disponibili; se è già presente una corrispondenza per quel nome essa verrà indicata come mostrato in figura 22.
Questa fase è fondamentale affinché il sistema proponga un mapping basato sulle relazioni create, ma non è strettamente richiesta in quanto esso può essere agevolmente eseguito anche manualmente in un secondo momento.

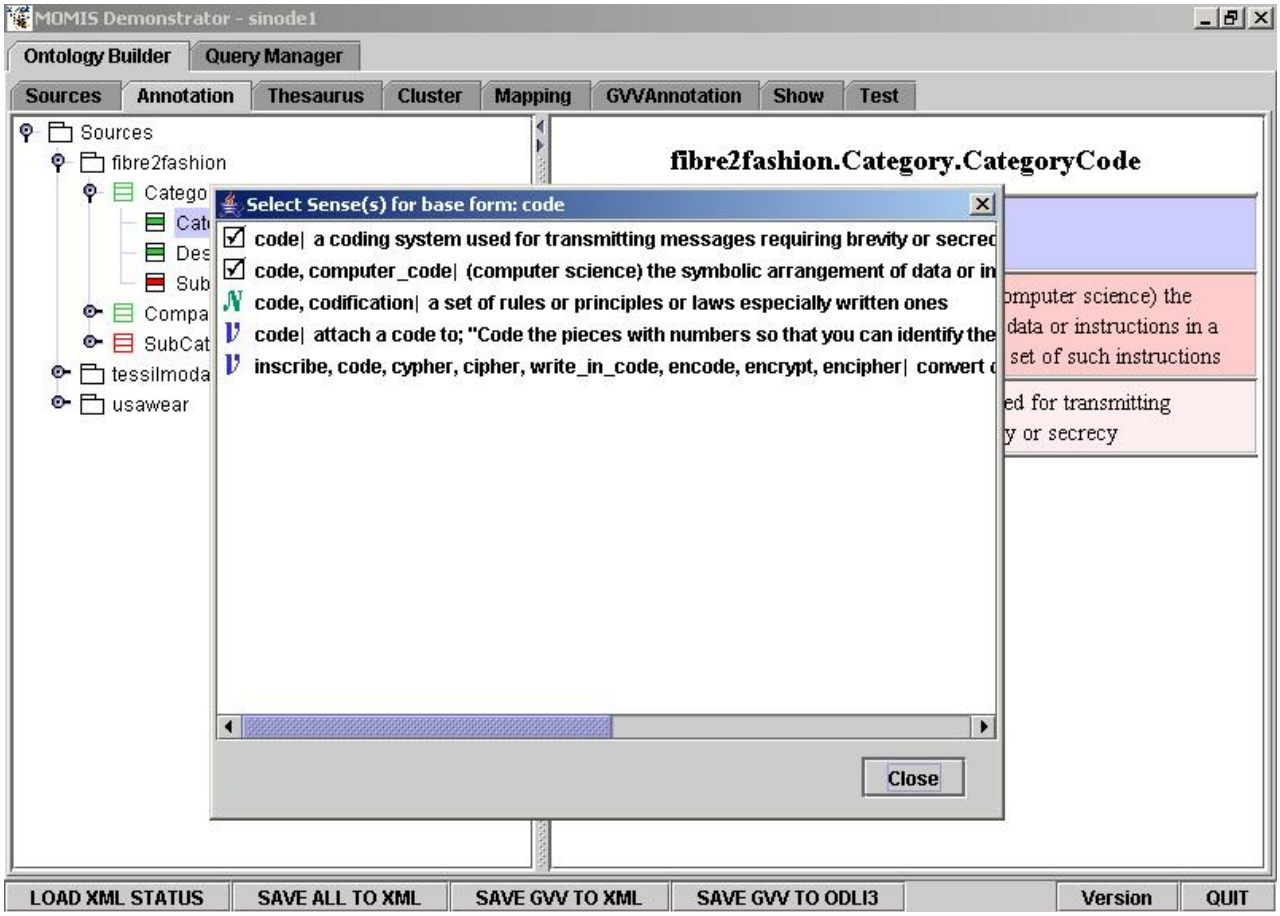



**Figura 21 – Assegnazione del significato**

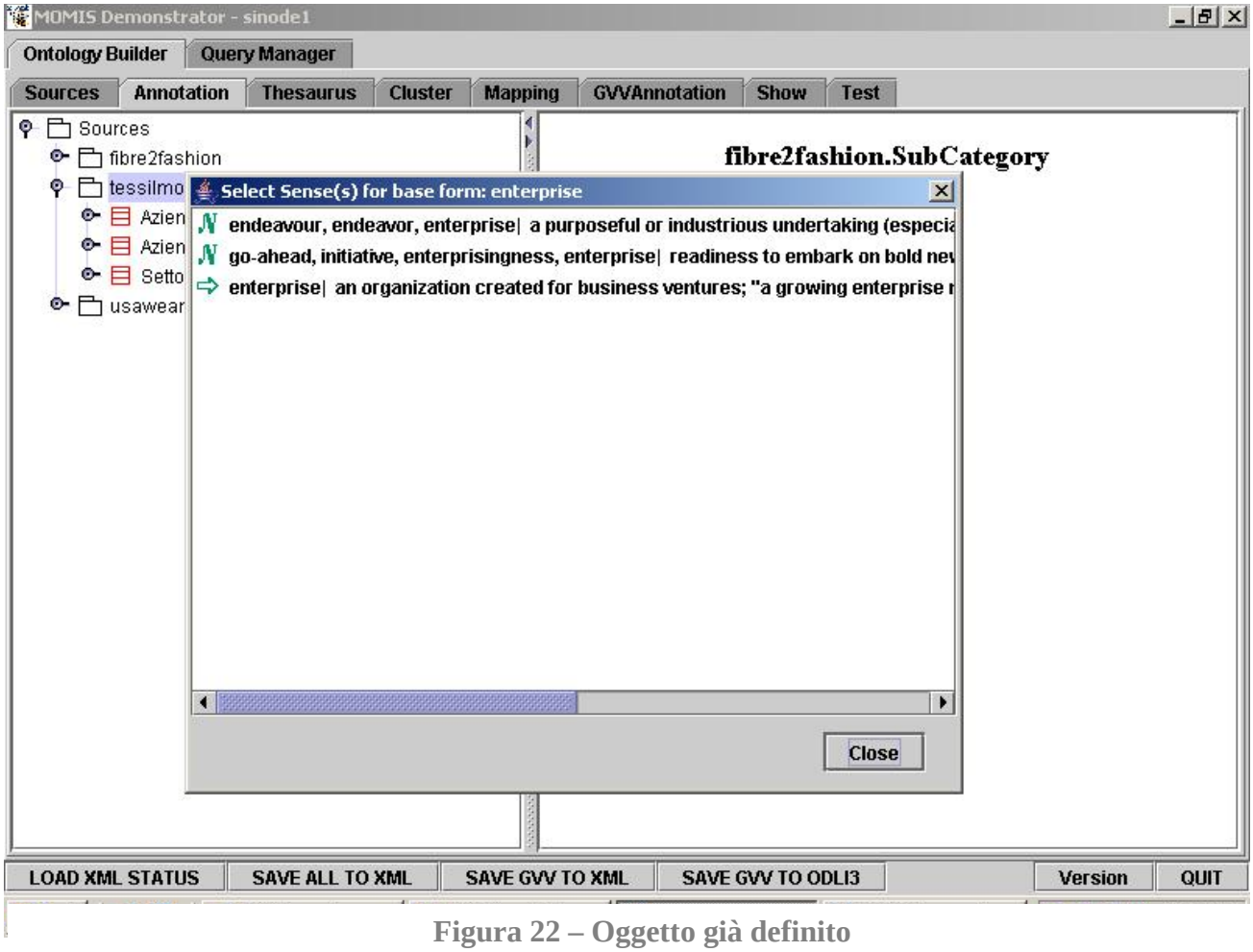



**Figura 22 – Oggetto già definito**

## 6.2.3 Definizione delle relazioni lessicali

Questa fase è necessaria solamente se si è provveduto ad eseguire totalmente od anche solo parzialmente lo step precedente; la sua funzione è quella di effettuare un processo di fusione tra i vari attributi e le varie entità a cui è stato assegnato lo stesso significato. Tutto ciò avviene in modo completamente automatico grazie all'apposito tasto "Lexicon-Derived Rels"

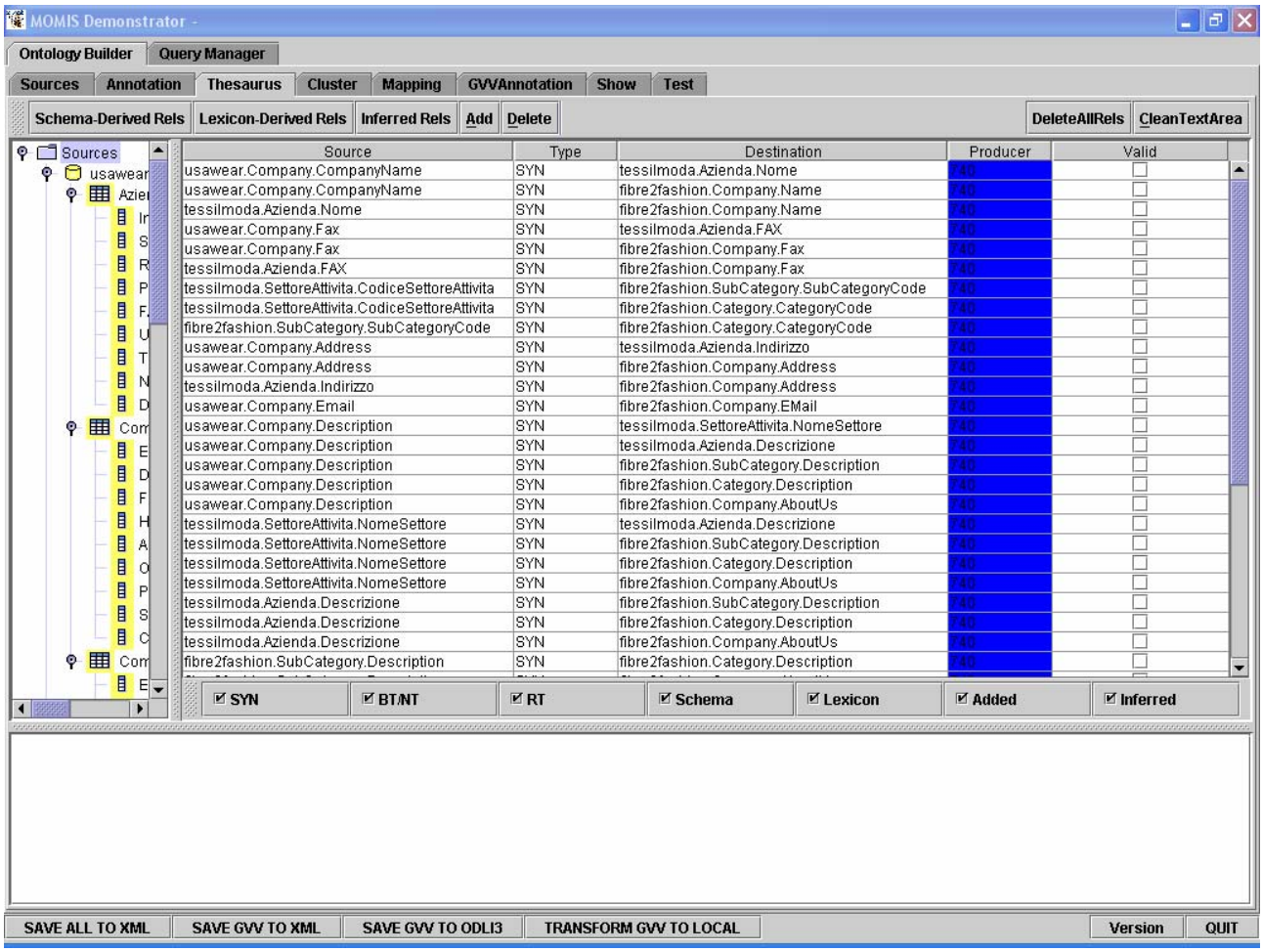



**Figura 23 – Definizione delle relazioni lessicali**

## 6.2.4 I Cluster

In questa stadio si definiscono i cluster; utilizzando gli appositi pulsanti è possibile definire le classi globali e mappare in esse le tabelle opportune.

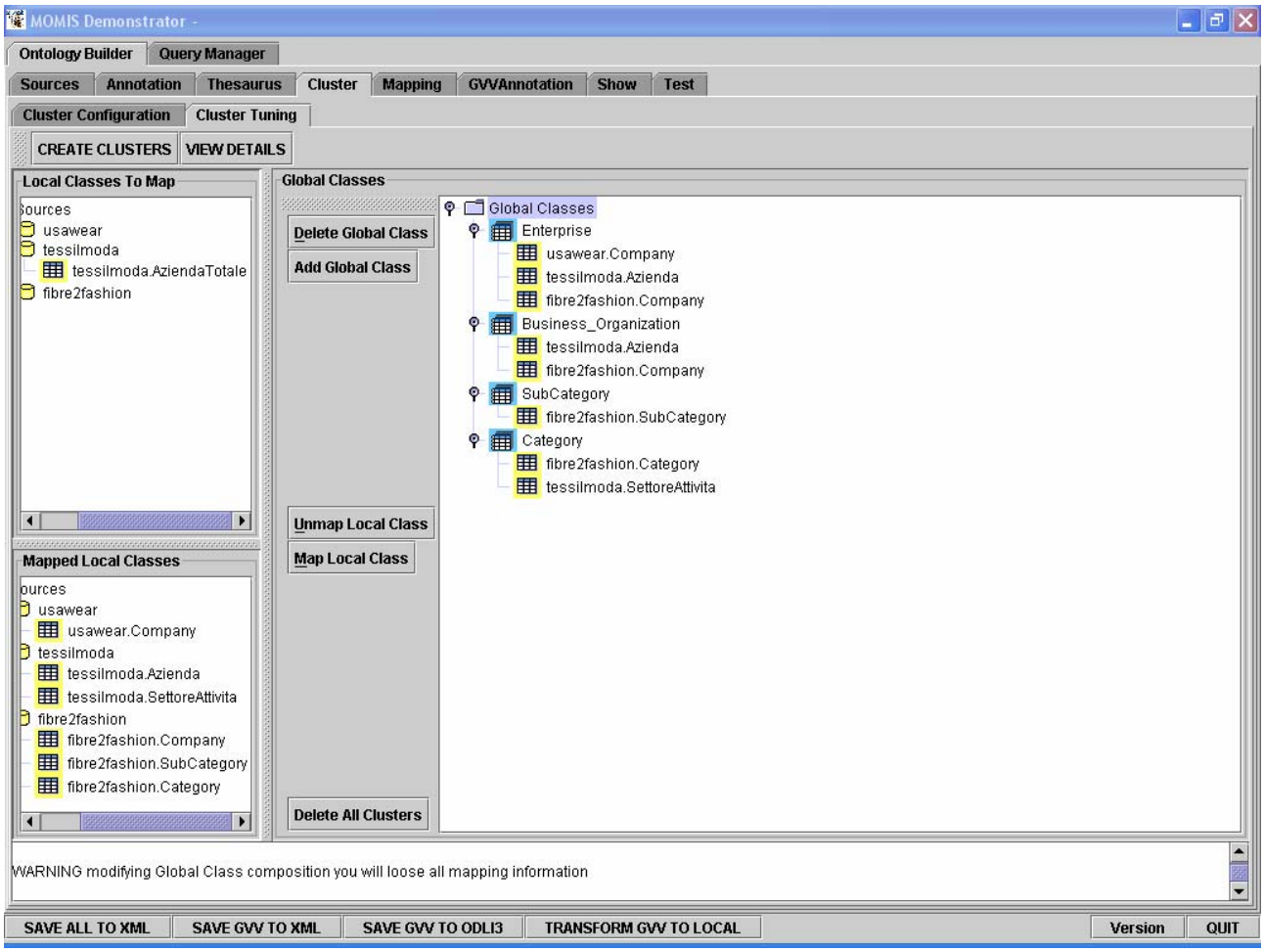



**Figura 24 – Realizzazione delle classi globali**

## 6.2.5 Il Mapping

In quest'ultima fase viene mostrato il mapping suggerito dal sistema; esso è facilmente modificabile trascinando i vari attributi da una classe all'altra. Nelle figure 25, 26, 27, 28 viene riportato il mapping, relativo alle varie classi, da me ottenuto che coincide con quello realizzato precedentemente dai ricercatori universitari.

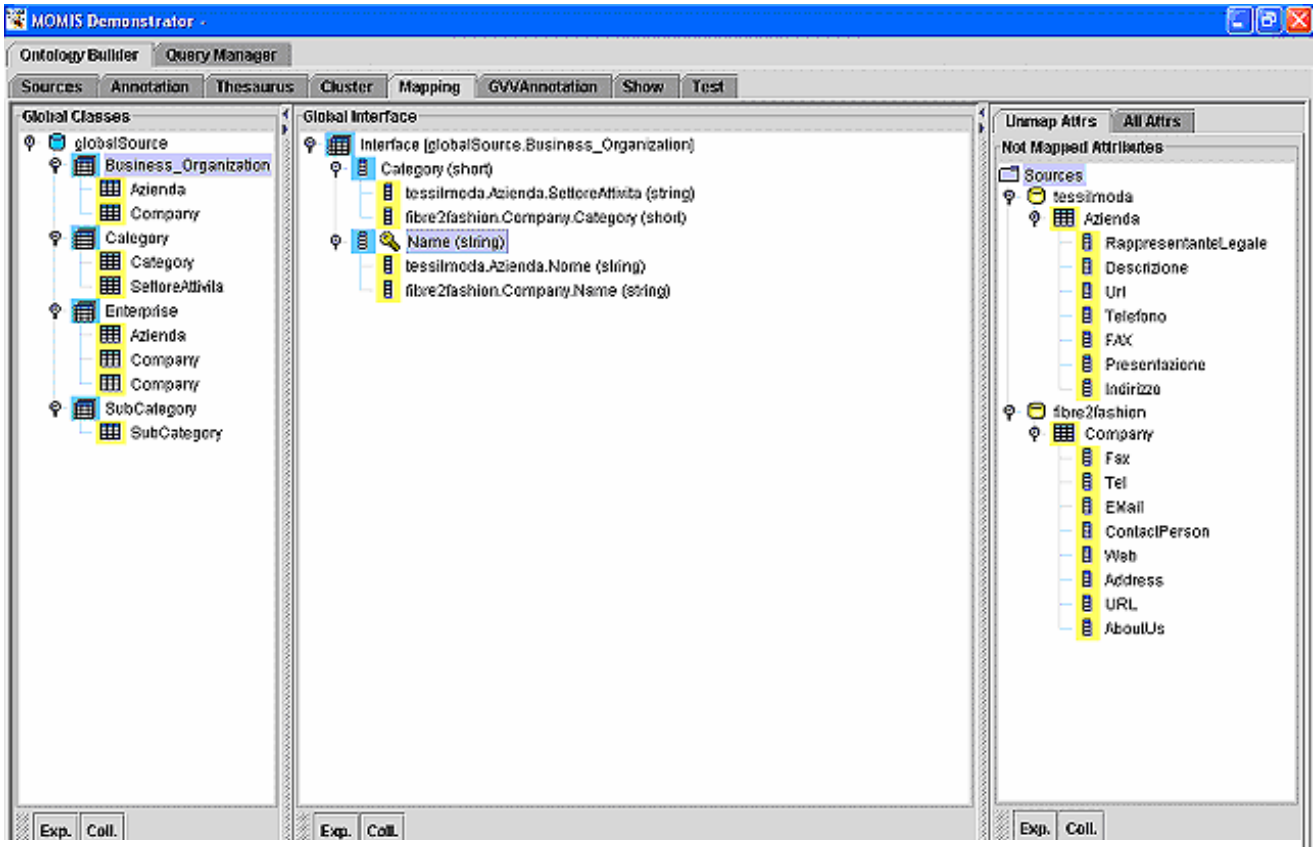



Figura 25 – Classe globale business_organization

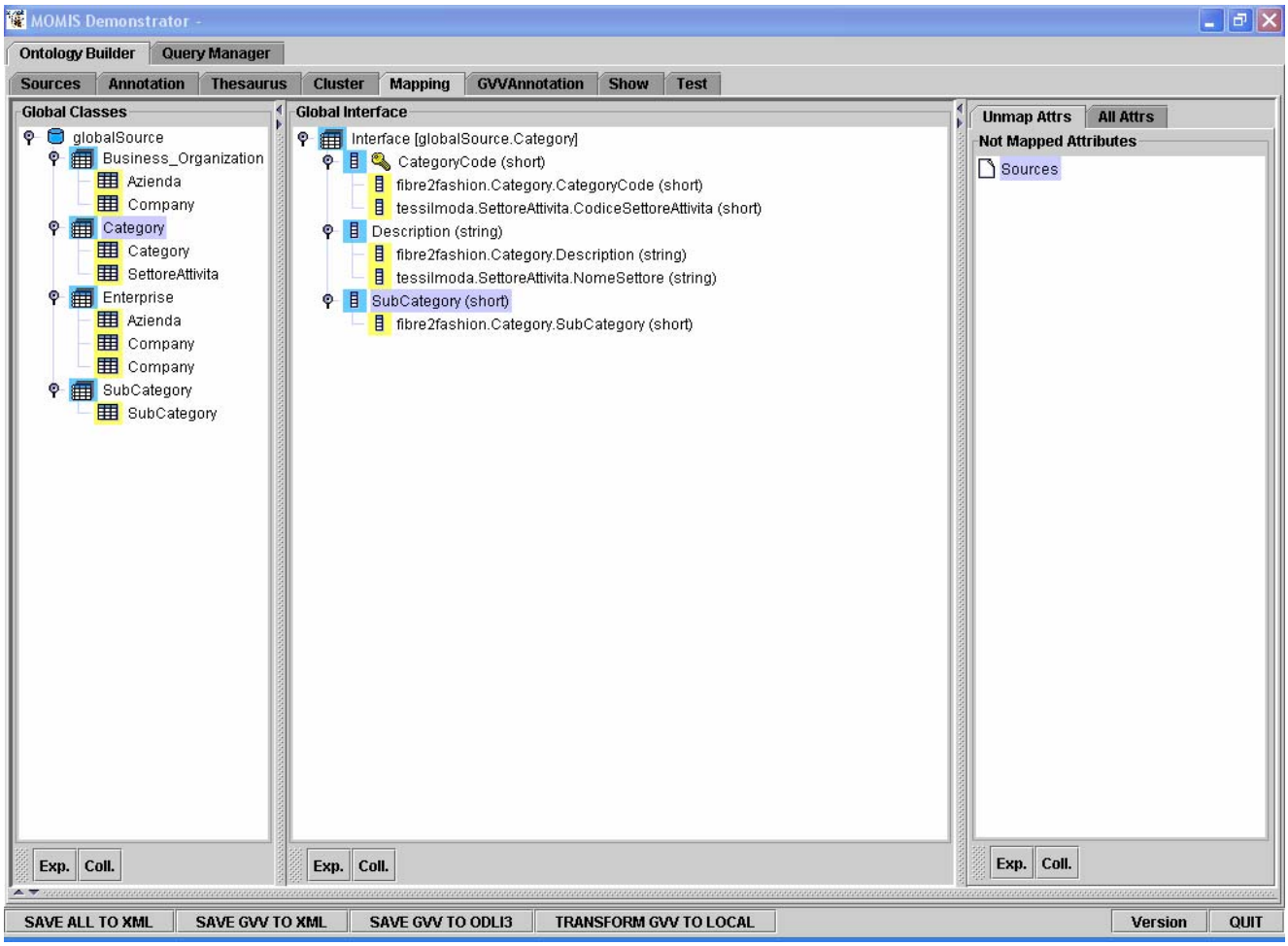



Figura 26 – Classe globale category

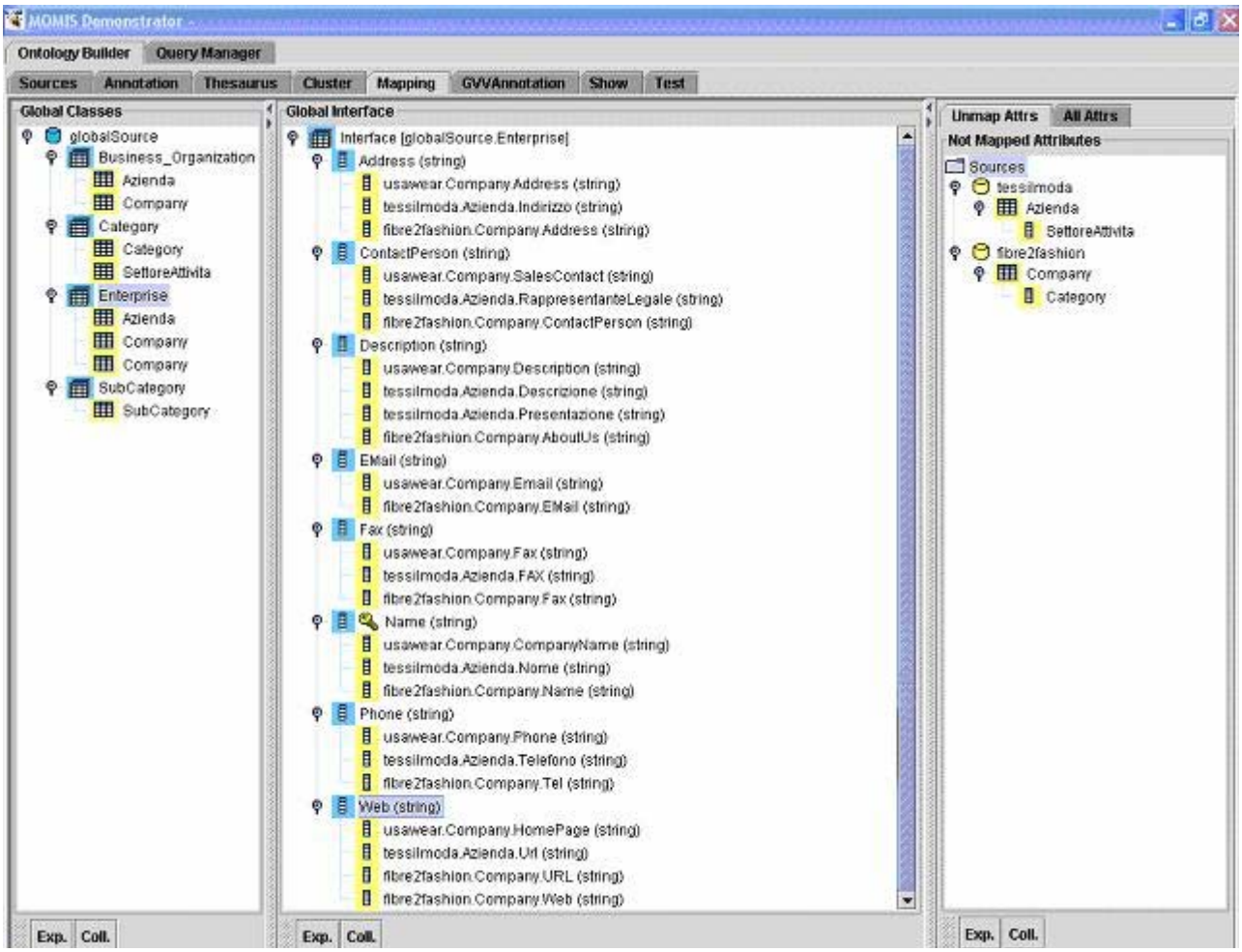



**Figura 27 – Classe globale enterprise**

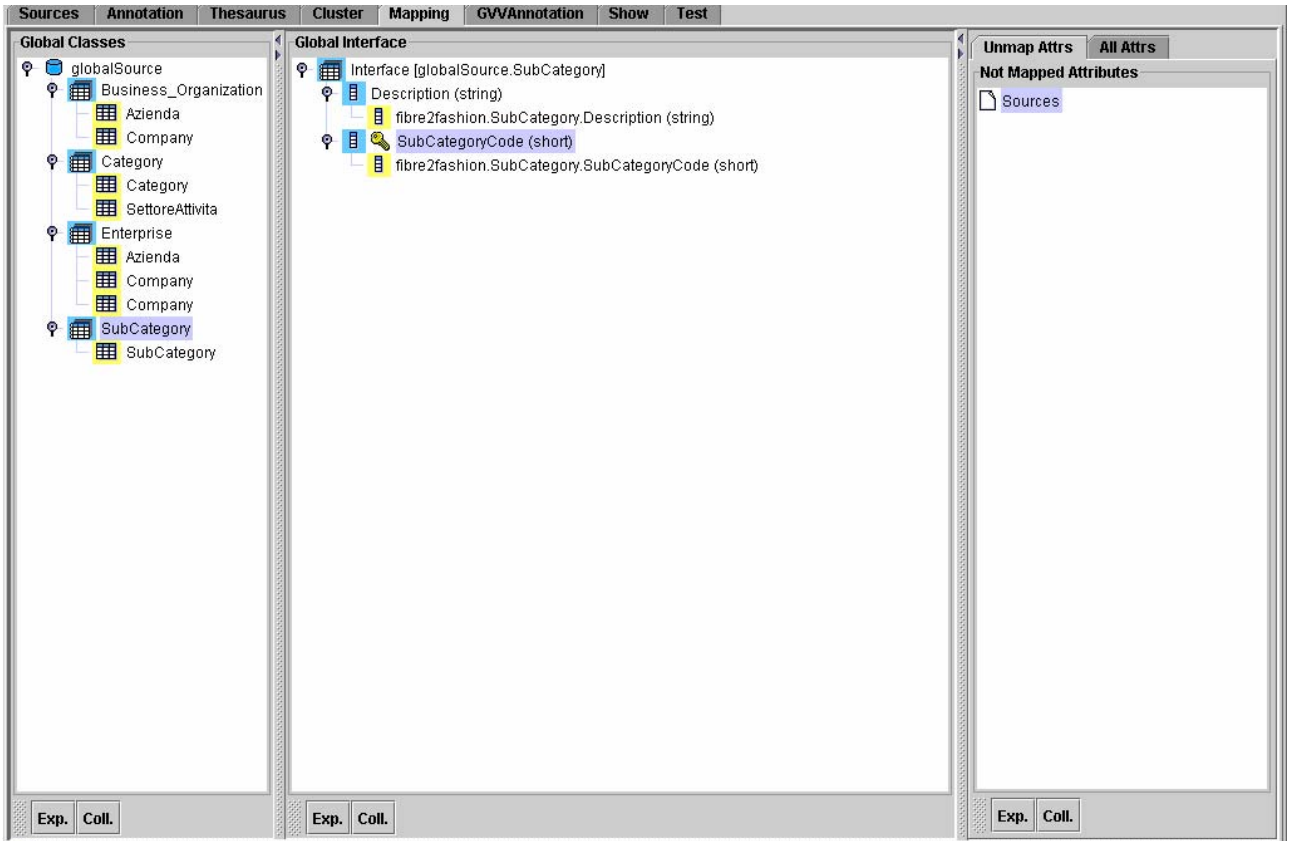



**Figura 28 – Classe globale subcategory**

L’intera struttura è stata sottoposta ad una serie di query atte a verificare l’effettiva integrazione dei dati e la tipologia di istruzioni SQL supportate. Dall’analisi compiuta è emerso che il set di comandi disponibile è piuttosto ristretto e rispecchia la seguente architettura:

SELECT [DISTINCT ] { * | nome campi }
FROM {nomi tabelle}
WHERE <criteri di selezione>
[GROUP BY]
[ORDER BY]

Con <criteri di selezione>:= nome campo-<operatore>-nome campo
<operatore>:= > , < , <>, = o LIKE.

È comunque importante sottolineare che il set di istruzioni verrà, in un futuro, sicuramente ampliato grazie al continuo sforzo da parte del DBGroup.

# Confrontro fra i due tool

Per poter effettuare un confronto il più attendibile e efficace possibile si è cercato di riprodurre in DB2 Information Integrator scenari già realizzati dal DBGroup dell'Università degli Studi di Modena e Reggio Emilia in ambiente MOMIS. L'analisi si articola in due fasi: la prima realizza l'integrazione di tre file XML, la seconda utilizza, al contrario, database di SQL Server.

## 7.1 Integrazione dei file XML

La struttura generale dei vari documenti viene riportata per permettere una migliore comprensione dell'ambiente esaminato.

- **ProntoComune**

```
<item>
   <Azienda>BAR RISTORANTE LA VECCHIA STAZIONE</Azienda>
   <Categoria>
     <Codice>71</Codice>
     <Descrizione>BAR, CAFFE', BIRRERIE, PANINOTECHE, PUBS</Descrizione>
   </Categoria>
   <Indirizzo>
     <Via>VIA VANDELLI, 234</Via>
     <CAP>41053</CAP>
     <Comune>POZZA DI MARANELLO</Comune>
   </Indirizzo>
   <Telefono>0536/948304</Telefono>
   <Fax>0536/948304</Fax>
   <Email>ka82@monrif.net</Email>
 </item>
```

- **IngroMarket**

```
<Azienda>
   <RagioneSociale>EVASION BY LA FULARISSIMA S.A.S.</RagioneSociale>
   <Attivita>Commercio</Attivita>
   <Merceologie>Valigeria e articoli da viaggio</Merceologie>
   <Indirizzo>
     <Via>Via Danubio, 47</Via>
     <Padiglione>Pad. E</Padiglione>
     <CAP>50010</CAP>
     <Localita>Osmannoro - Sesto Fiorentino</Localita>
     <Provincia>FI</Provincia>
   </Indirizzo>
   <Telefono>055-315355</Telefono>
   <Fax>055-316898</Fax>
   <Email>evasion@ingromarket.it</Email>
   <Orari>lu-ve 8,30-18,00</Orari>
   <Titolari>Joelle Salliou</Titolari>
 </Azienda>
```

- **SubFor**

```
<Azienda>
        <CodiceAzienda>1003868</CodiceAzienda>
        <Settore>
                <CodiceSettore>subforMeccanicaCodiceSettore11</CodiceSettore>
                <Descrizione>Meccanica</Descrizione>
        </Settore>
        <Nome>FILOSTAMP S.r.l.</Nome>
        <Indirizzo>
                <Via>Via Val Della Torre 48</Via>
                <CAP>10040</CAP>
                <Localita>CASELETTE</Localita>
                <Provincia>TORINO</Provincia>
                <Regione>Piemonte</Regione>
                <Telefono>011/9688108</Telefono>
                <Fax>011/9688730</Fax>
                <Email>filostamp@excite.it</Email>
        </Indirizzo>
        <Contatti>- DONATO Silvio , Amministratore, Lingue parlate: Inglese</Contatti>
        <Contatti>- SINA Giovanna , Segretaria, Lingue parlate: Francese</Contatti>
        <AnnoInizio>ANNO DI INIZIO DELL'ATTIVITÀ: 1982</AnnoInizio>
        <Addetti>ADDETTI: 14 addetti , di cui 9 operai</Addetti>
        <Turni>TURNI DI LAVORO: 1</Turni>
        <Fatturato>FATTURATO:(consuntivi e/o previsioni) 1448 mila Euro nel 2001</Fatturato>
        <SerieProduzione>SERIE DI PRODUZIONE:  - Grande serie</SerieProduzione>
        <FatturatoSubfornitura>FATTURATO SUBFORNITURA: 100% </FatturatoSubfornitura>
        <OffertaSubfornitura>OFFERTA DI SUBFORNITURA:</OffertaSubfornitura>
        <MaterialeLavorato>Acciaio inossidabile</MaterialeLavorato>
        <MercatiInternazionali>FATTURATO EXPORT: 6%</MercatiInternazionali>
        <Committenza>I principali committenti sono:</Committenza>
        <Committenza>Grandi imprese o gruppi</Committenza>
        <Decentramento>L'azienda affida a ditte esterne l'esecuzione di lavorazioni o fasi del
        processo produttivo</Decentramento>
        <Decentramento>L'azienda e interessata a ricercare nuovi subfornitori in
        regione</Decentramento>
        <Decentramento>L'azienda e interessata a ricercare nuovi subfornitori altrove: hinterland
        milanese</Decentramento>
        <SboccoPrincipale>SBOCCO PRINCIPALE: Automobili</SboccoPrincipale>
        <Tecnologie>- Sistema di controllo e programmazione delle attivita
        produttive</Tecnologie>
        <Tecnologie>- Robot di manipolazione</Tecnologie>
</Azienda>
```

Come si può notare i dati contenuti nei file sono altamente eterogenei, questo richiederà un maggiore sforzo durante il processo d'integrazione.

### *7.1.1 Mapping dei dati*

Per ottenere una struttura complessiva contenente tutte le informazioni è necessario realizzare un mapping per i dati comuni.
In MOMIS, come già visto, questo viene fatto semi-automaticamente dal sistema. In DB2 Information Integrator la situazione è completamente differente, se da una parte non vi è la necessità di portare a termine il laborioso processo di attribuzione dei significati ai vari attributi, dall'altra non è presente nessun tipo di funzionalità di mapping diretto. Questo implica la necessità

di realizzare tale procedura manualmente attraverso la stesura di codice SQL che realizzi la creazione di viste le quali, attraverso delle UNION, raggruppino insieme le informazioni necessarie. Nelle figure 29 30 31 viene mostrato il mapping eseguito in MOMIS. Successivamente sono riportate le query realizzate per ottenere la stessa struttura nel software IBM.

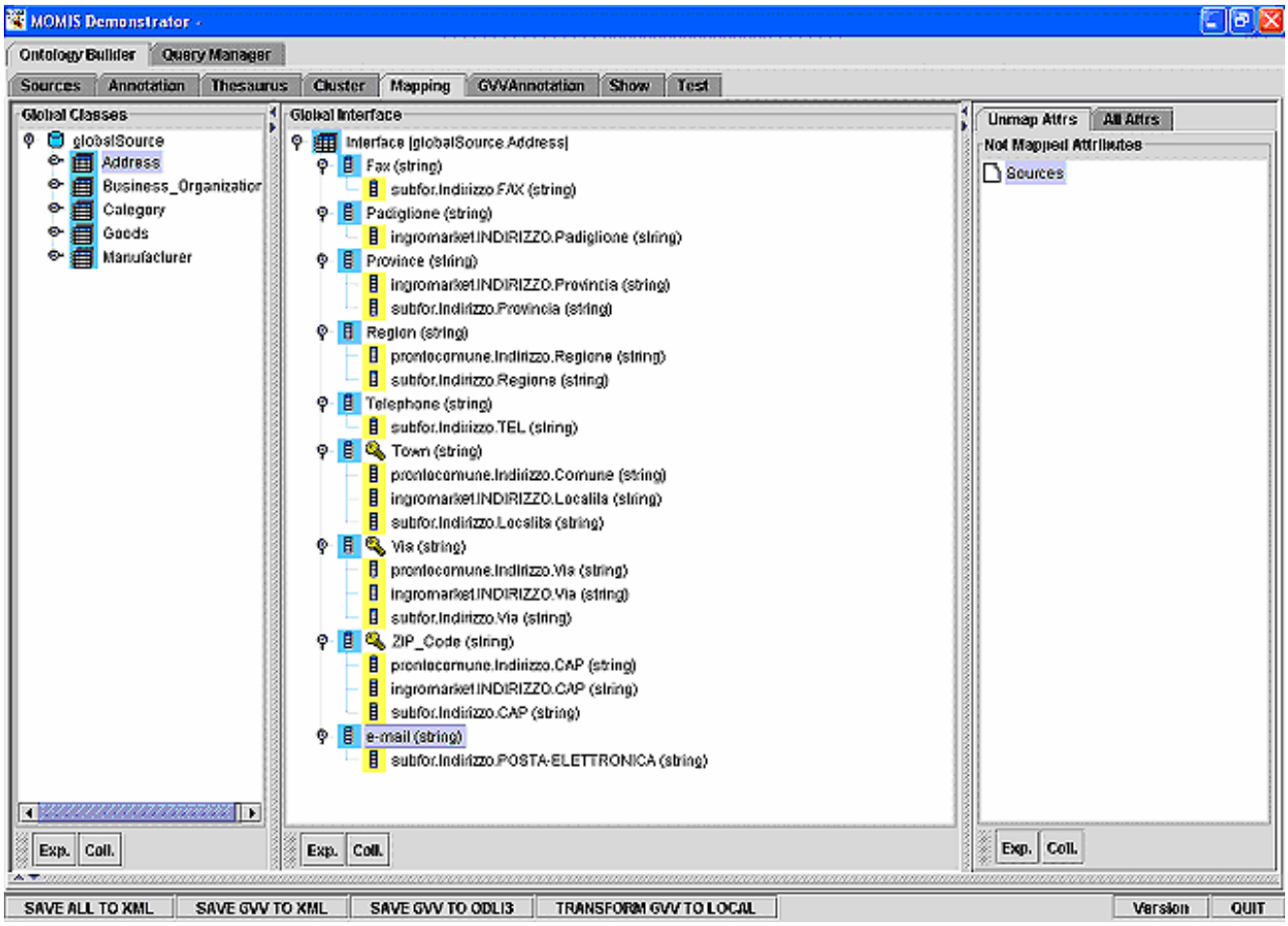



**Figura 29 – Classe globale address**

```
create view address as
        select   nome, indirizzo_via as via, indirizzo_cap as cap, indirizzo_localita as localita,
                 indirizzo_provincia as provincia, ' ' as regione, indirizzo_padiglione as padiglione
        from     azienda_im
        union
        select   nome, indirizzo_via as via, indirizzo_cap as cap, indirizzo_localita as localita,  ' ' as
                 provincia, indirizzo_regione as regione, ' ' as padiglione
        from     item_pc
        union
        select   nome, indirizzo_via as via, indirizzo_cap as cap, indirizzo_localita as localita,
                 indirizzo_provincia as provincia, indirizzo_regione as regione, ' ' as padiglione
        from     azienda_sf
```

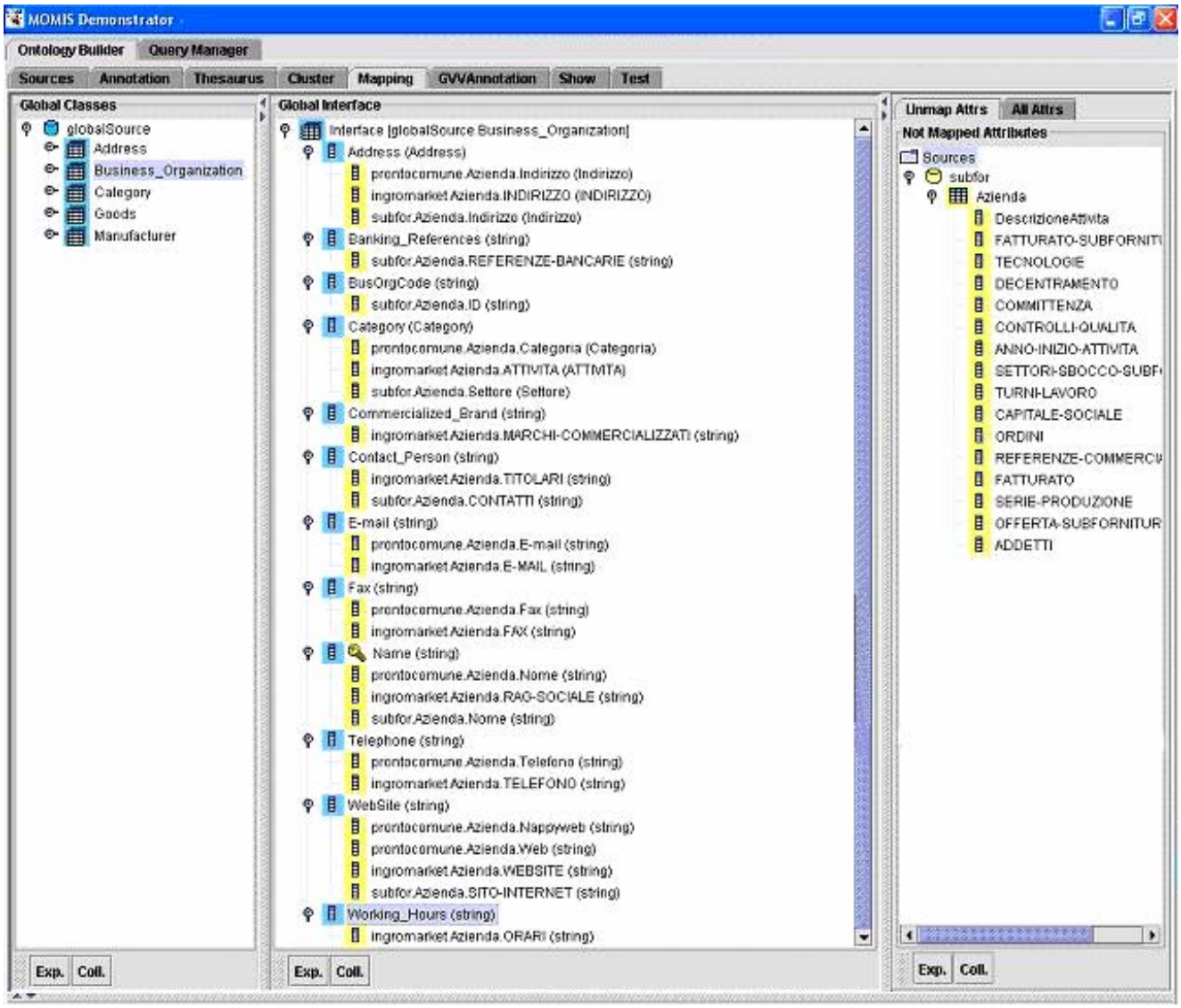



**Figura 30 – Classe globale business_organization**

```
create view     business_organization as
        select  nome, ' ' as codiceSettore, settore_descrizione as descrizioneSettore, telefono,
                indirizzo_fax as fax, indirizzo_email as email, web, ' ' as nappyweb, orari, contatti as
                contatti, marchicommercializzati
        from    azienda_im
        union
        select  nome, settore_codice as codiceSettore, settore_descrizione as descrizioneSettore,
                telefono, indirizzo_fax as fax, indirizzo_email as email, web, nappyweb, ' ' as orari, '
                ' as contatti, ' ' as marchicommercializzati
        from    item_pc
        union
        select  nome, settore_codice as codiceSettore, settore_descrizione as descrizioneSettore,
                a.telefono as telefono, indirizzo_fax as fax, indirizzo_email as email, ' ' as web,' ' as
                nappyweb, ' ' as orari, c.contatti as contatti, ' ' as marchicommercializzati
        from    indirizzo a
                join    azienda_sf z on a.azienda_sf_id=z.azienda_sf_id
                join    azienda_azienda c on c.azienda_sf_id=z.azienda_sf_id
```

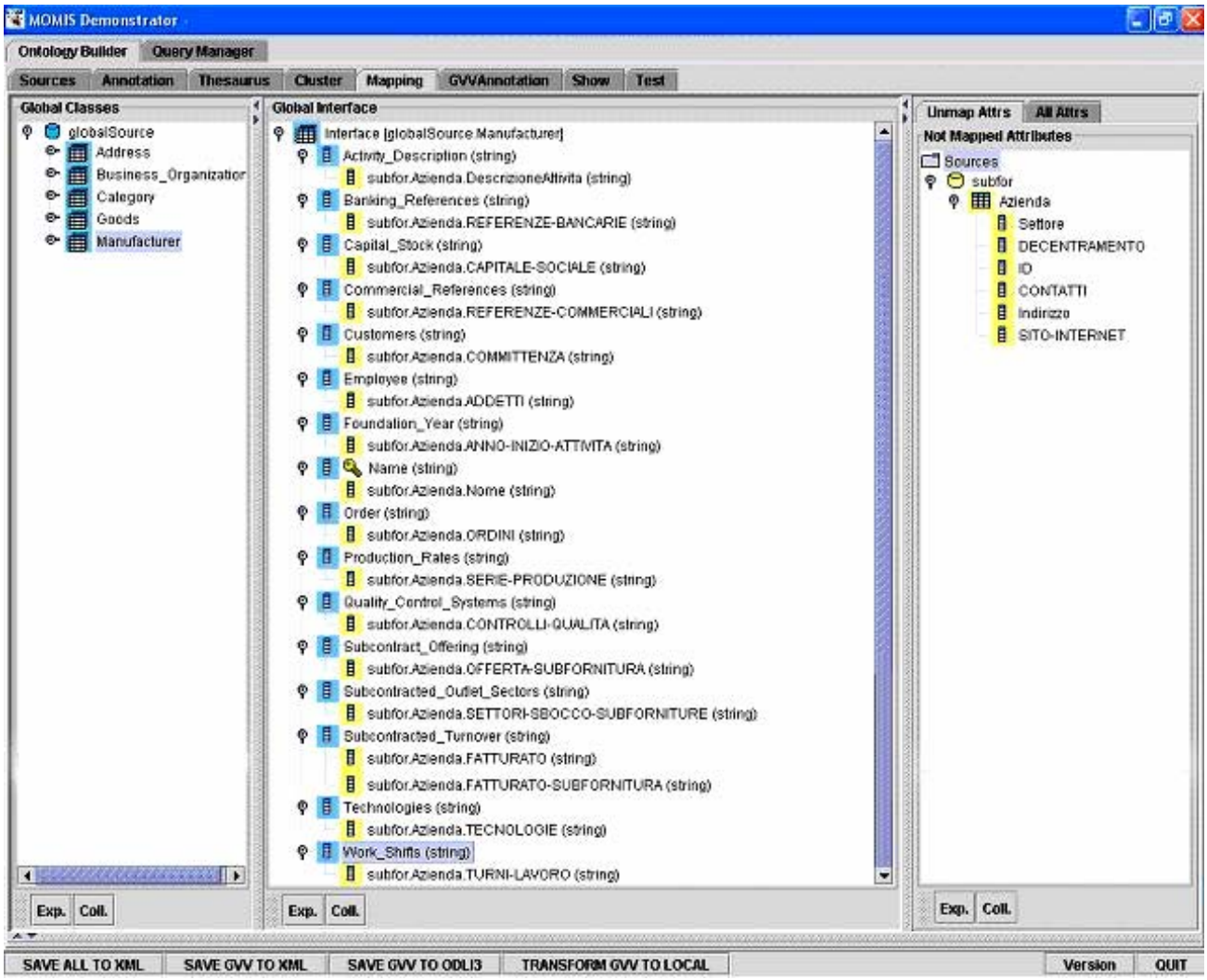



**Figura 31– Classe globale manufacturer**

```
create view manufacturer as
        select   nome, descrizioneattivita, annoinizio, addetti, turni, fatturato, serieproduzione,
                 fatturatosubfornitura, offertasubfornitura, settorisboccoforniture, capitalesociale,
                 c.committenza as commitenza, t.tecnologie as tecnologie
        from     azienda_azienda4
                 join     azienda_sf a on a.azienda_sf_id=t.azienda_sf_id
                 join     azienda_azienda2 c on c.azienda_sf_id=t.azienda_sf_id
```

Come si può notare nei frame a sinistra delle figure esistono due classi globali, Category e Goods, che non sono state riprodotte in ambiente DB2 Information Integrator; il motivo di questa scelta è dovuto al fatto che le informazioni presenti in esse erano già incluse nella vista *business_organization* e che una loro ulteriore definizione sarebbe risultata ridondante. Un'altra differenza tra i due scenari è che, mentre MOMIS mappa i campi *telefono* e *fax* relativi al file SubForn in *address* e quelli attinenti a ProntoComune e IngroMarket in *business_organization*, si è preferito optare per una soluzione che raggruppasse tali attributi in un'unica locazione: *business_organization*.
È importante, inoltre, sottolineare che occorre eseguire un processo di alterazione dei nickaname in modo che essi contengano campi con la stessa tipologia di dati in modo da permettere la realizzazione delle classi globali.

## 7.1.2 Capacità di DB2 Information Integrator

Questa sezione analizza la struttura ricreata nel software IBM. Lo scopo di questo paragrafo è quello di riportare varie considerazioni riguardanti le sue potenzialità. Per realizzare ciò si è sottoposta tale struttura ad una serie d'interrogazioni contenenti le funzioni base di SQL, esse sono tutte query piuttosto semplici ed elementari e, in alcuni casi, di scarso interesse pratico, ma il loro scopo non è tanto quello di rappresentare potenti strumenti d'interrogazione bensì di permetterne un'analisi il più possibile esauriente.

### 7.1.2.1 Funzioni matematiche e query innestate

Tutte le funzioni aritmetiche come AVG, MAX, MIN, SUM… sono previste e ammesse dal software. Non essendoci attributi di tipo integer non è stato possibile provarle direttamente sui file in esame, esse sono però state testate su viste inerenti ad altri nickname dimostrando la loro piena compatibilità con strutture d'integrazione.
È stato inoltre verificato che il sistema supporta query innestate. Come esempio viene riportata una query che seleziona i nomi di tutte le aziende che non hanno sede a Bologna realizzata prima con la clausola NOT IN e, in un secondo momento, con NOT EXISTS.

```
select  b_o1.nome
from    business_organization as b_o1
where   nome not in (  select  b_o2.nome
                       from    business_organization as b_o2
                       join    address as a on b_o2.nome=a.nome
                       where   a.provincia='bologna')
```

```
select  b_o1.nome
from    business_organization as b_o1
where   not exists (   select  *
                       from    business_organization as b_o2
                       join    address as a on b_o2.nome=a.nome
                       where   a.provincia='bologna')
```

Il risultato ottenuto è, ovviamente, lo stesso in entrambi i casi ed è mostrato in figura 32.

### 7.1.2.2 Group by

Per verificare la clausola GROUP BY è stata realizzata una query che restituisce la lista delle aziende e delle località presso cui esse hanno sede "raggruppate" per località.

```
select          e.nome, a.localita
from            business_organization as b_o
join            address as a on b_o.nome=a.nome
group by        a.localita, e.nome
```

Il risultato è mostrato in figura 33.

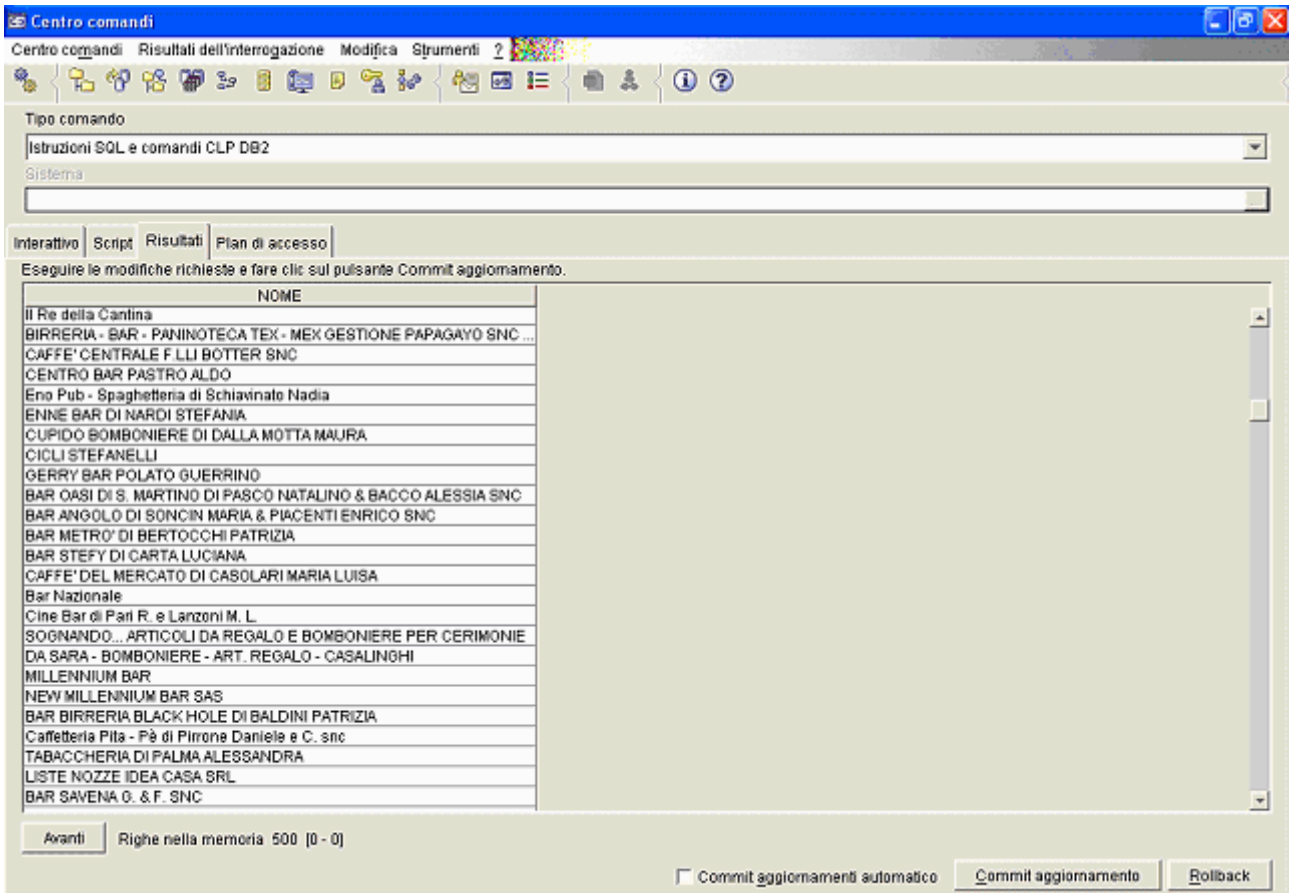



Figura 32 – Query innestate

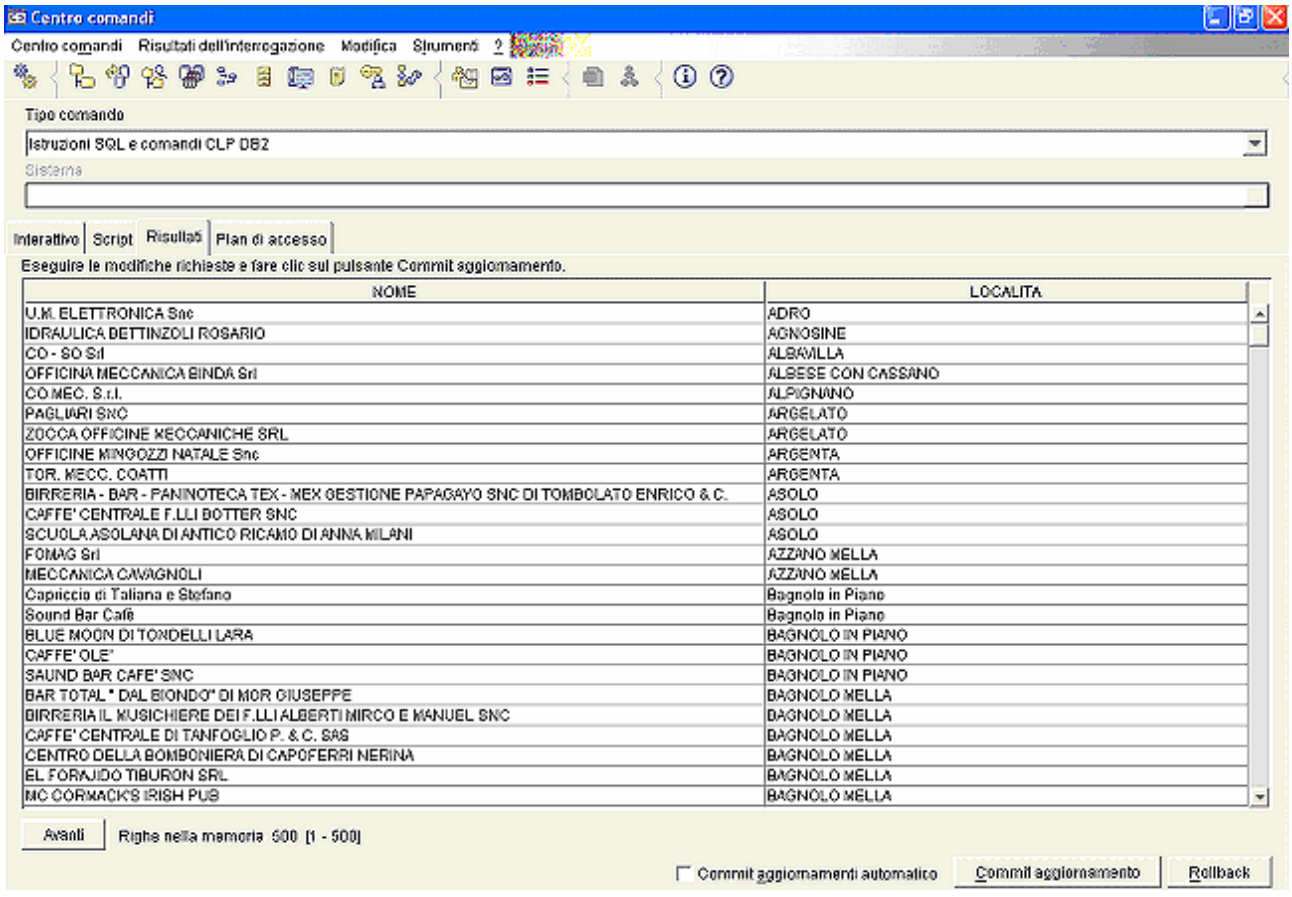



Figura 33 – Group by

## *7.1.2.3 Order by*

L'analisi ha portato alla luce che non è possibile ordinare, direttamente nella loro definizione, viste che raggruppano informazioni appartenenti a più nickname. Se si vuole ottenere un elenco ordinato, ad esempio in base al nome, delle aziende presenti nel sistema è necessario realizzare una apposita query che selezioni tutti i campi di *business_organization* e che contenga la clausola ORDER BY.

```
select      *
from        business_organization
order by    nome
```

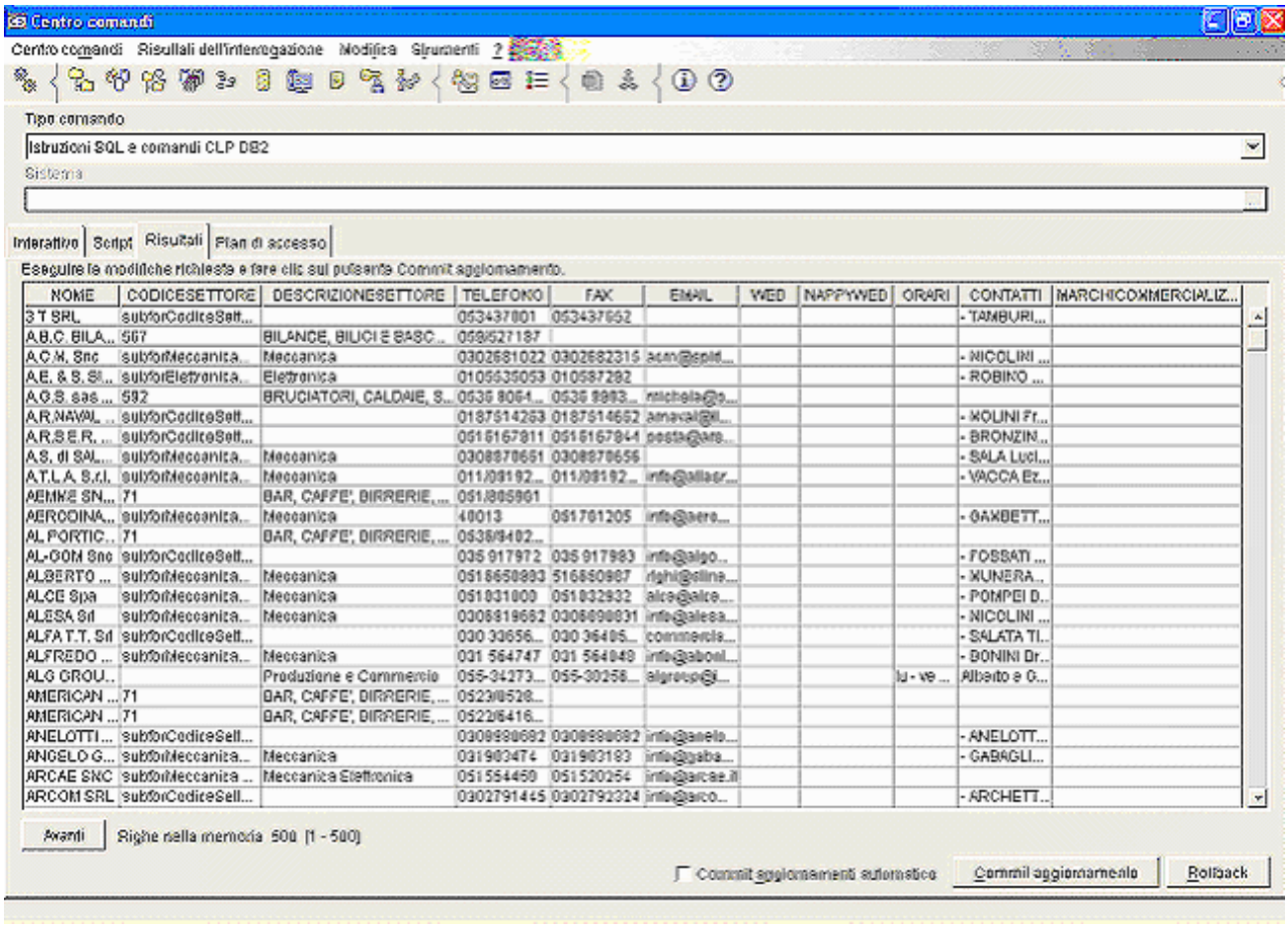



**Figura 34 – Order by**

## *7.1.2.4 Count(*)*

Per verificare la COUNT(*) è stata creata una query che contasse il numero di aziende alle quali corrisponde un numero di fax

```
select   count(*) as 'Numero ditte con fax'
from     business_organization
where    fax is not null.
```

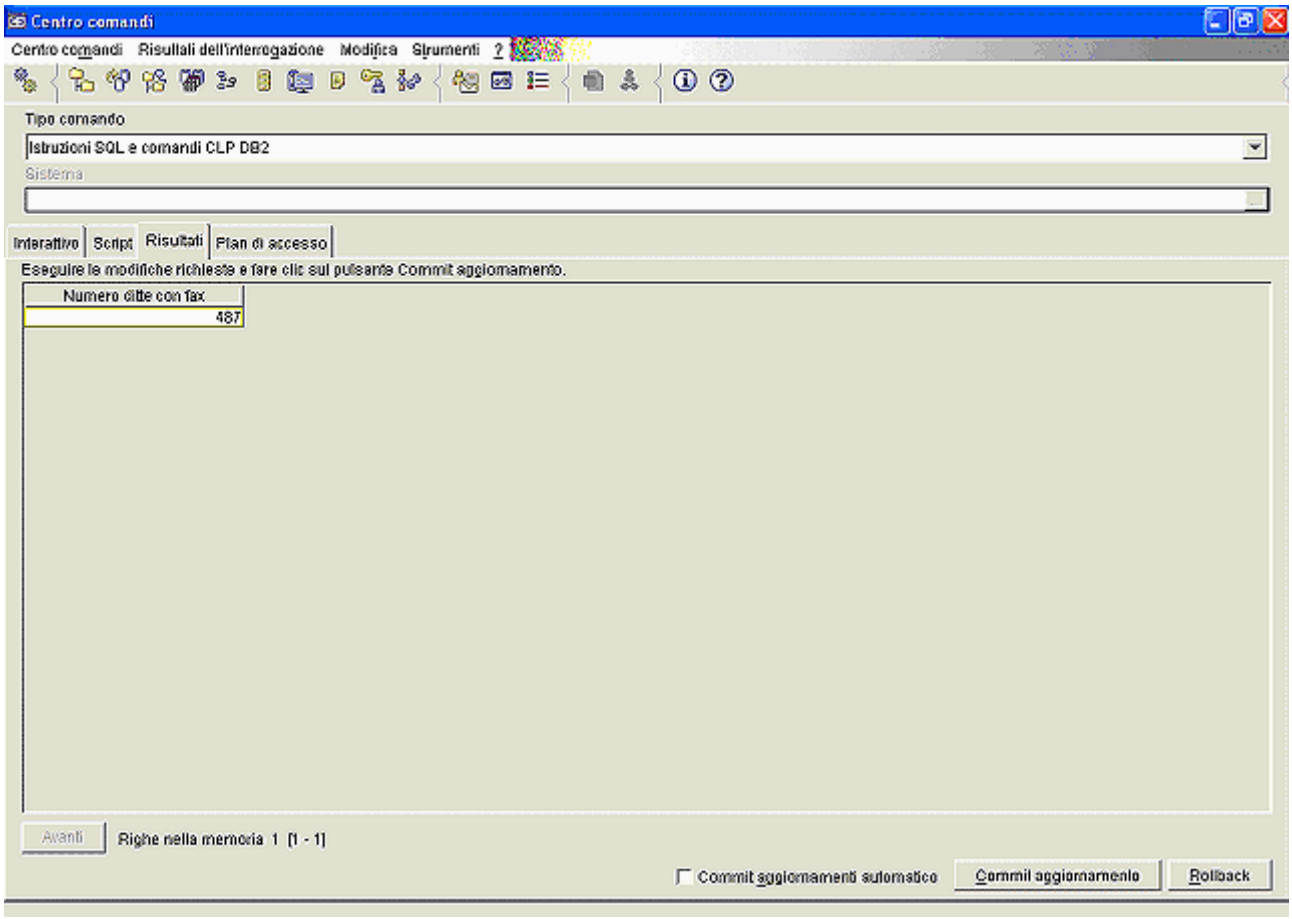



**Figura 35– Count(*)**

### *7.1.2.5 Having*

Supponiamo, ora, di voler visualizzare le province in cui hanno sede più di tre aziende; per realizzare ciò è necessario ricorrere al costrutto HAVING come qui di seguito riportato.

```
select    a.provincia, count(*) as "Numero di aziende per località"
from      address as a
join      business_organization as b_o on a.nome=b_o.nome
group by a.provincia
having    count(*)>3
order by  2
```

### *7.1.2.6 Like*

La seguente query visualizza i nomi e i numeri di telefono di tutte le ditte che contengono, in *DescrizioneSettore*, la parola caffe ricorrendo a LIKE. Anche in questo caso è stata mostrata la possibilità di utilizzare questo costrutto fondamentale di SQL.

```
select    nome, DescrizioneSettore, Telefono
from      business_organization
where     DescrizioneSettore like '%CAFFE%'
```

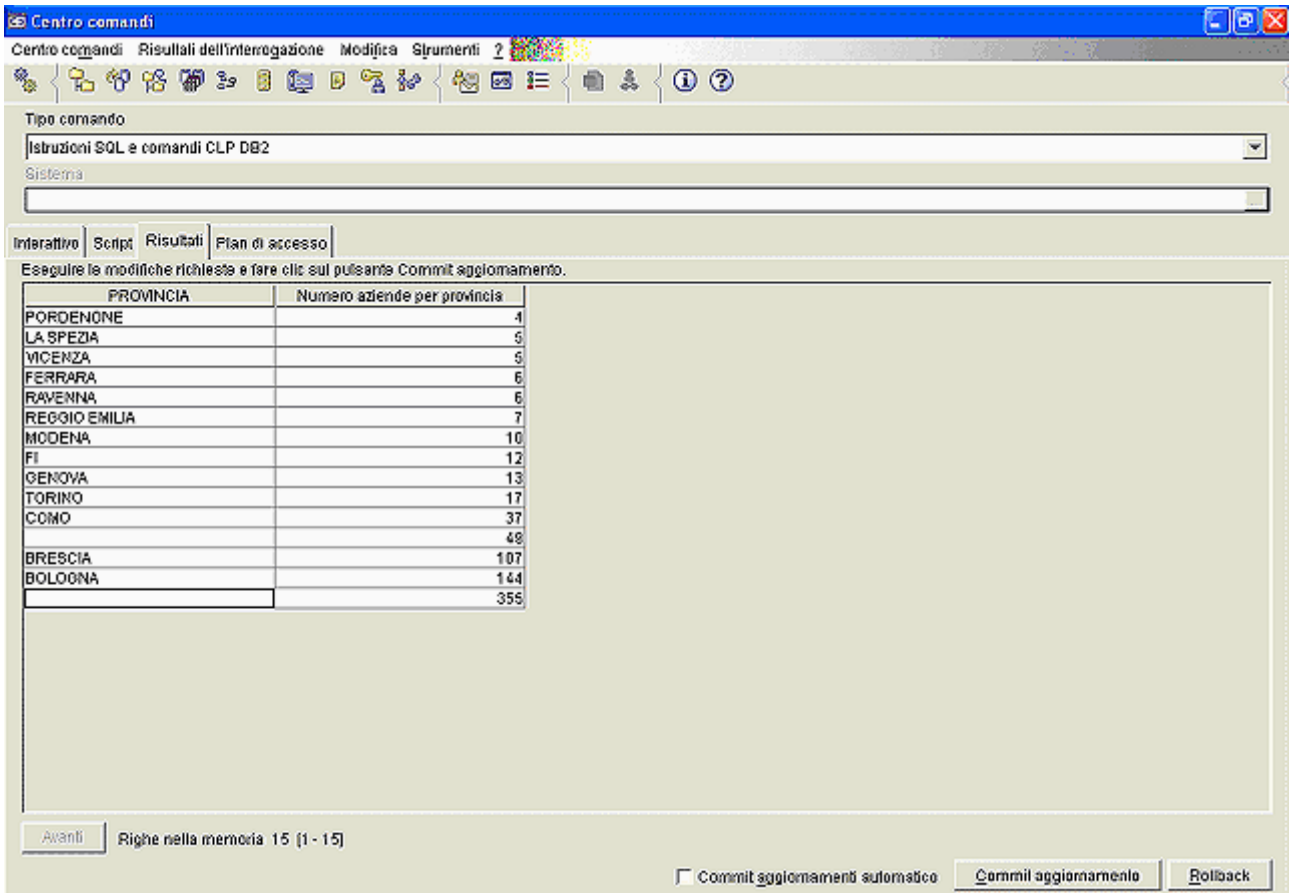



**Figura 36 – Having**

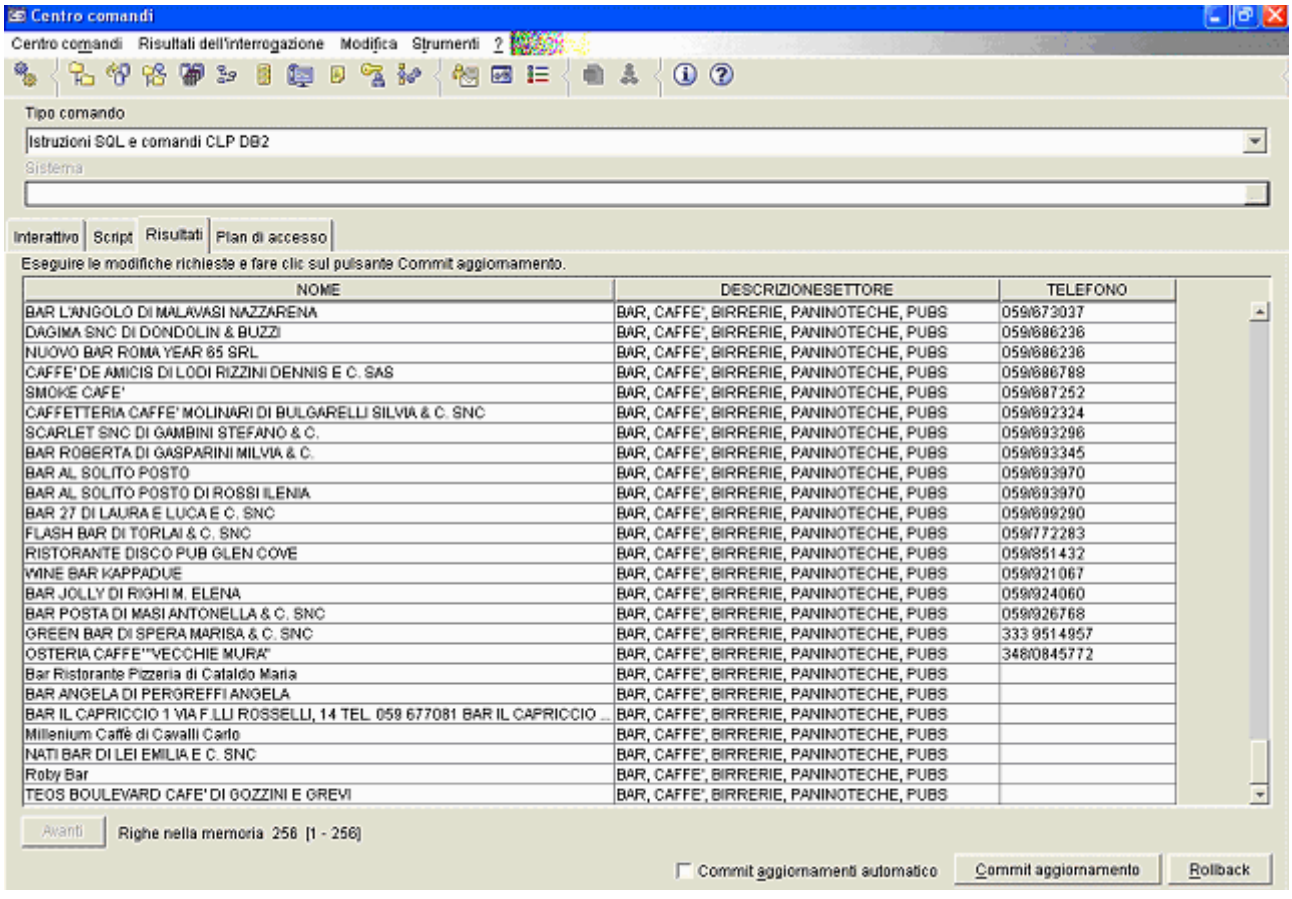



**Figura 37 – Like**

### 7.1.2.7 Indici

Il linguaggio SQL non permette la creazione di indici che hanno come oggetto viste, DB2 Information Integrator continua a mantenere questa limitazione consentendone, comunque, la realizzazione per i nickname come qui di seguito riportato.

```
create index    indice
on              item_pc (nome)
specification only
```

È importante sottolineare che è strettamente necessario includere, nella definizione di indici riguardanti nickname, l'opzione *specification only* affinché non siano prodotti errori.

### 7.1.2.8 Trigger

Il sistema federato non prevede la creazione di trigger che operano sugli oggetti che lo compongono, non è stato dunque possibile definirli per i nickname e per le viste ad essi inerenti. Permane comunque la possibilità di creare questi strumenti per tabelle di DB2 come indicato nel seguente esempio. Esso mostra un trigger che, per ogni nuova tupla inserita in *clienti* ne aggiunge una a *indirizzi* inserendo nel campo *id* l'identificativo del nuovo cliente e settando *null* gli altri attributi.

```
Create Trigger          inserimento
after insert on         clienti
referencing new as      nuovo
for each row mode DB2SQL
insert into indirizzi values (nuovo.id, null, null)
```

### 7.1.2.9 Record duplicati

In questo paragrafo si analizza un caso particolare che potrebbe verificarsi quando, come in questa situazione, si integrano file tra loro differenti. Sono stati esaminati tre casi limite: il primo prevede la presenza di due record esattamente identici all'interno di due documenti distinti, il secondo rappresenta la situazione in cui tali record contengano valori diversi all'interno di un solo campo, nel caso specifico *telefono*, ed il terzo dove le tuple duplicate si trovano nello stesso file.

Caso a) Le informazioni vengono riprodotte in entrambi i nickname corrispondenti ai due documenti. Nella vista, che rappresenta la classe globale, il record non viene duplicato, le informazioni vengono riprodotte una sola volta.

```
select  *
from    business_organization
where   nome like 'TEQUILA%'
```

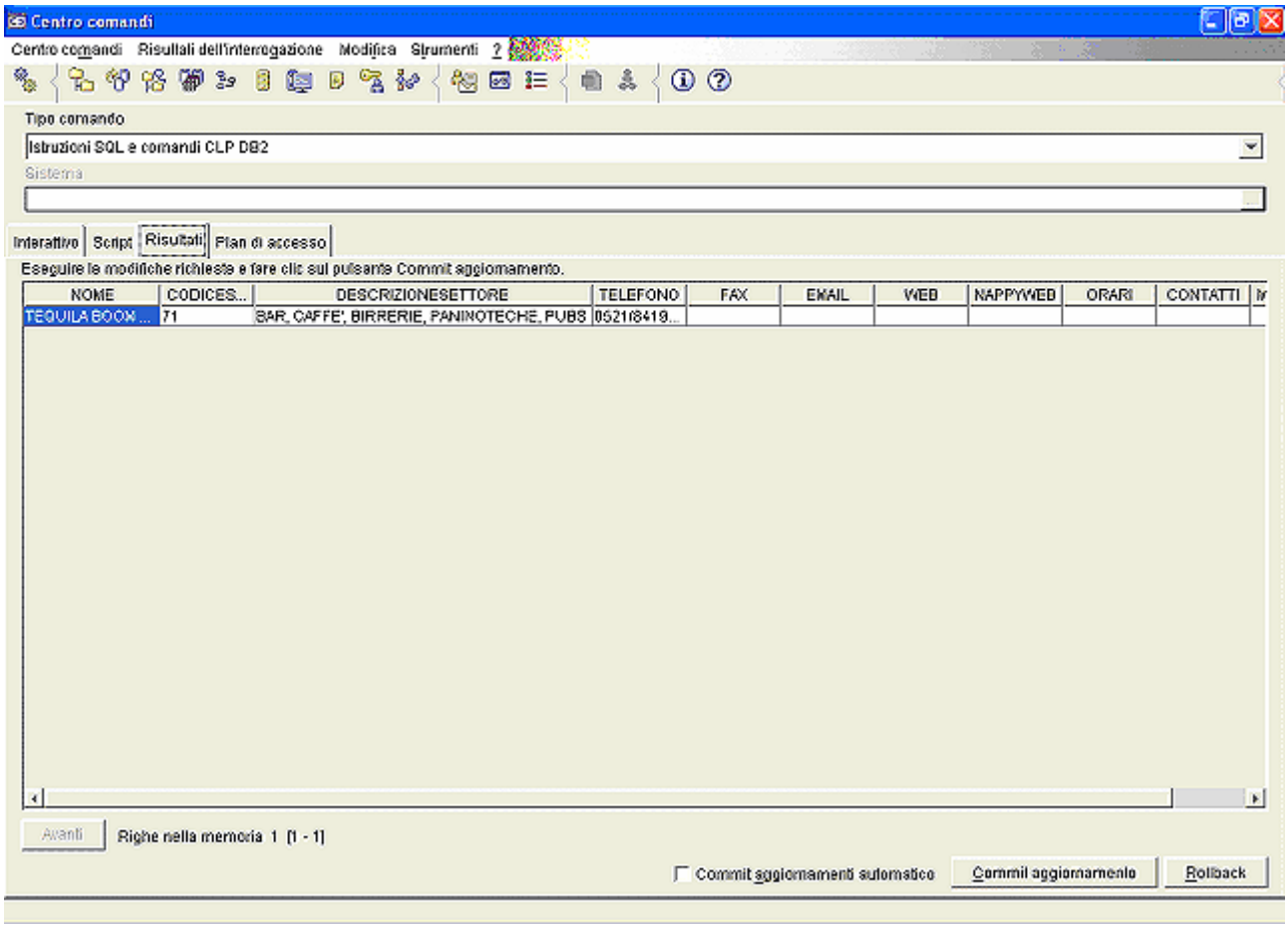



**Figura 38 – business_organization con due record identici**

Questo è verificato solo nel caso particolare in cui i record duplicati abbiano la stessa struttura e contengano gli stessi dati. Nel caso in cui in una delle due tuple sia definito anche un solo attributo non presente nell'altra, entrambi i record verranno visualizzati anche se il valore del campo in questione è *null*.

Caso b) Ovviamente nella situazione in cui i due record presentino almeno un campo con valori differenti risulta ovvio che verranno entrambi inseriti nella classe (vedere figura 39).

```
select   *
from    business_organization
where  nome like 'TEQUILA%'
```

Caso c) All'interno del file ProntoComune  sono stati definiti due record esattamente identici. Dopo la definizione del nickname relativo a tale documento si nota che esso contiene entrambe le tuple. Quando però si va a creare la vista *business_organization* si nota che, come nel primo caso, non vi sono problemi di ridondanza come mostrato in figura 38.

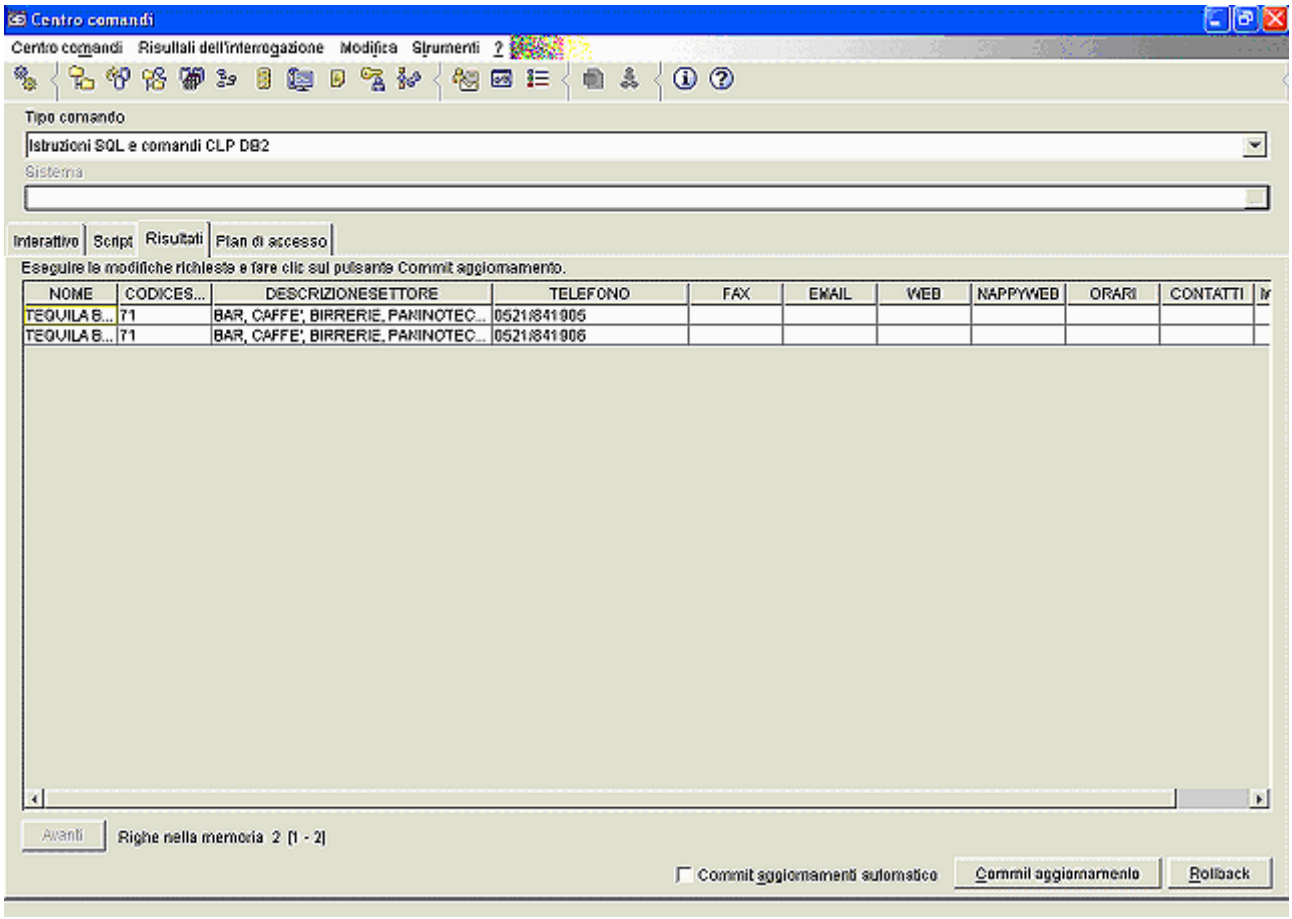


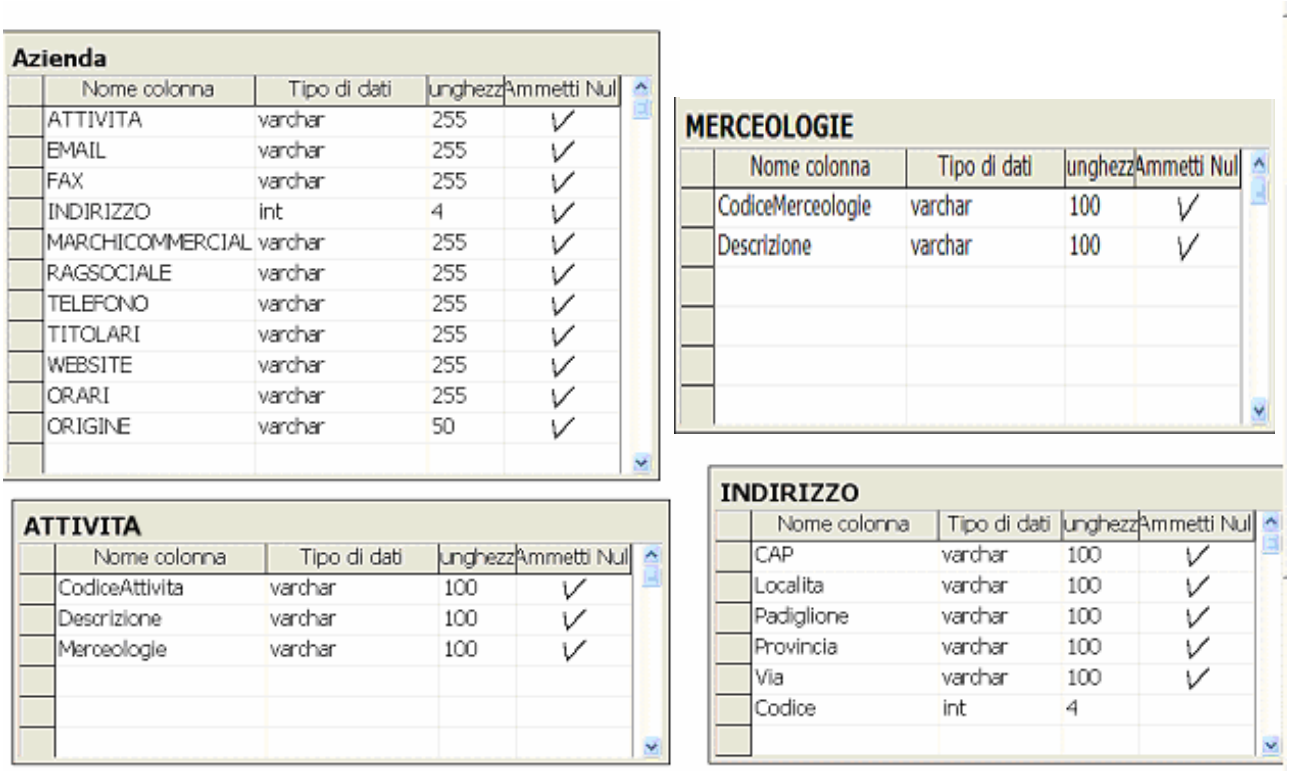

**Figura 39 – business_organization con due record differenti**

## 7.2 Integrazione dei database di SQL Server

Questo paragrafo riguarda il processo d'integrazione di sei database. Le loro strutture sono riportate qui di seguito; si noti che tre di essi rappresentano la trasposizione in modello relazionale dei file XML precedentemente analizzati mentre i restanti corrispondono a quelli già analizzati nel capitolo precedente (figure 16 17 18).

All'interno dell'ambiente MOMIS sono presenti due livelli d'integrazione, uno che raggruppa i database in due strutture (*sinode1, sinode2*), ognuna delle quali composta da tre database, l'altro che integra tra loro queste due astrazioni.

Riprodurre questo scenario in DB2 Information Integrator risulta, almeno per la realizzazione del primo livello, molto simile a quanto finora esposto; si tratta infatti di realizzare viste che seguano il mapping di MOMIS per poi crearne delle ulteriori che inglobino le precedenti.



**Azienda**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| ATTIVITA | varchar | 255 | ✓ |
| EMAIL | varchar | 255 | ✓ |
| FAX | varchar | 255 | ✓ |
| INDIRIZZO | int | 4 | ✓ |
| MARCHICOMMERCIAL | varchar | 255 | ✓ |
| RAGSOCIALE | varchar | 255 | ✓ |
| TELEFONO | varchar | 255 | ✓ |
| TITOLARI | varchar | 255 | ✓ |
| WEBSITE | varchar | 255 | ✓ |
| ORARI | varchar | 255 | ✓ |
| ORIGINE | varchar | 50 | ✓ |

**MERCEOLOGIE**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CodiceMerceologie | varchar | 100 | ✓ |
| Descrizione | varchar | 100 | ✓ |

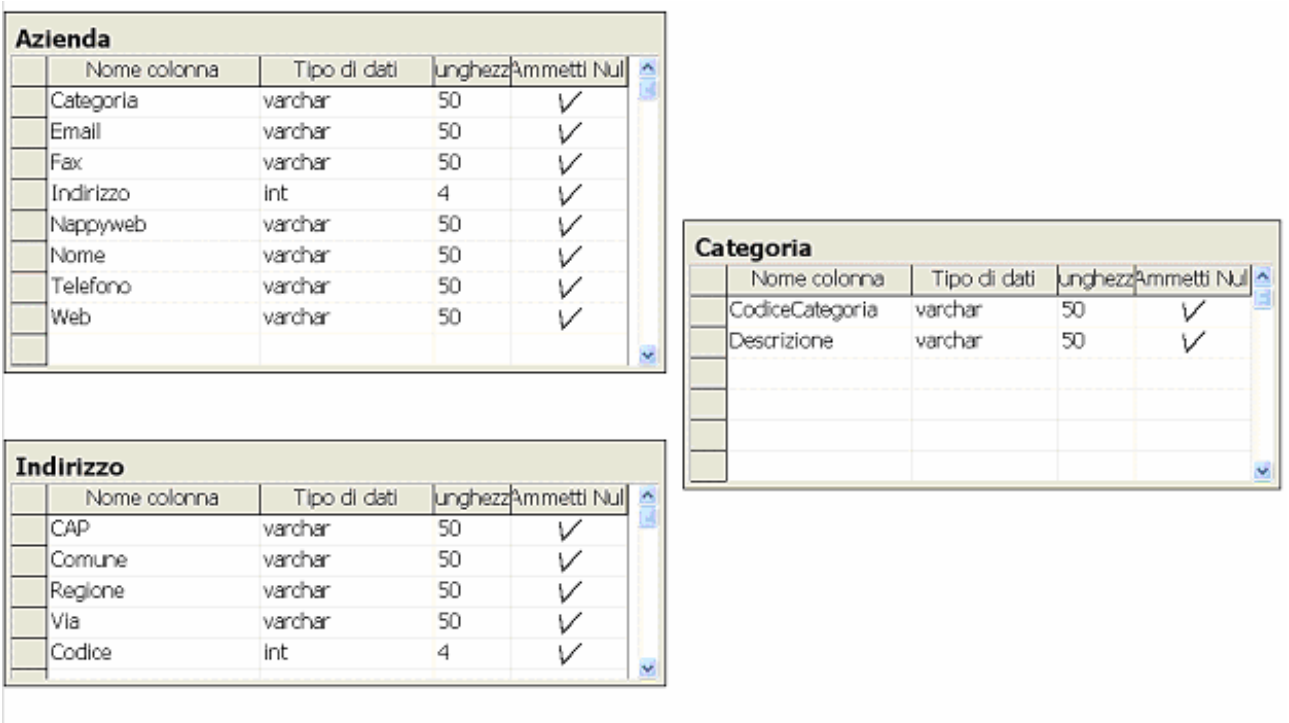
**ATTIVITA**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CodiceAttivita | varchar | 100 | ✓ |
| Descrizione | varchar | 100 | ✓ |
| Merceologie | varchar | 100 | ✓ |

**INDIRIZZO**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CAP | varchar | 100 | ✓ |
| Localita | varchar | 100 | ✓ |
| Padiglione | varchar | 100 | ✓ |
| Provincia | varchar | 100 | ✓ |
| Via | varchar | 100 | ✓ |
| Codice | int | 4 | |

**Figura 40 – Struttura db Ingromarket**



**Azienda**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| Categoria | varchar | 50 | ✓ |
| Email | varchar | 50 | ✓ |
| Fax | varchar | 50 | ✓ |
| Indirizzo | int | 4 | ✓ |
| Nappyweb | varchar | 50 | ✓ |
| Nome | varchar | 50 | ✓ |
| Telefono | varchar | 50 | ✓ |
| Web | varchar | 50 | ✓ |

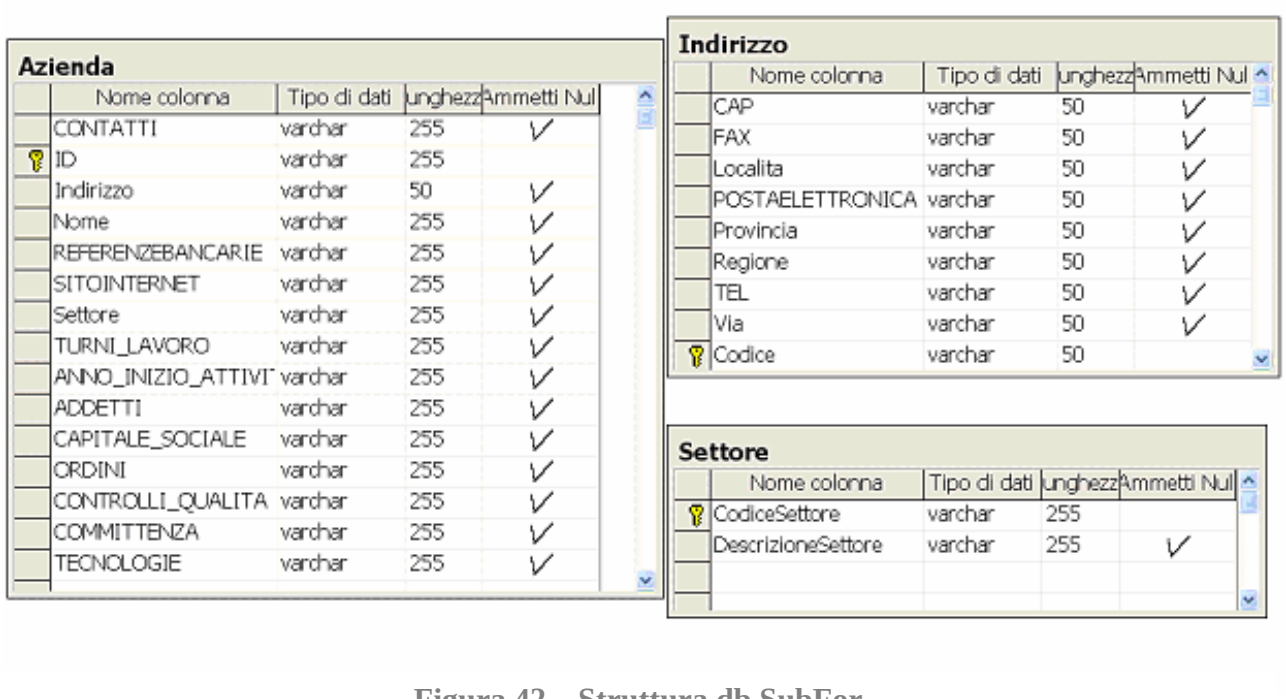
**Categoria**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CodiceCategoria | varchar | 50 | ✓ |
| Descrizione | varchar | 50 | ✓ |

**Indirizzo**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CAP | varchar | 50 | ✓ |
| Comune | varchar | 50 | ✓ |
| Regione | varchar | 50 | ✓ |
| Via | varchar | 50 | ✓ |
| Codice | int | 4 | ✓ |

**Figura 41 – Struttura db ProntoComune**



**Azienda**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CONTATTI | varchar | 255 | ✓ |
| ID | varchar | 255 | |
| Indirizzo | varchar | 50 | ✓ |
| Nome | varchar | 255 | ✓ |
| REFERENZEBANCARIE | varchar | 255 | ✓ |
| SITOINTERNET | varchar | 255 | ✓ |
| Settore | varchar | 255 | ✓ |
| TURNI_LAVORO | varchar | 255 | ✓ |
| ANNO_INIZIO_ATTIVI | varchar | 255 | ✓ |
| ADDETTI | varchar | 255 | ✓ |
| CAPITALE_SOCIALE | varchar | 255 | ✓ |
| ORDINI | varchar | 255 | ✓ |
| CONTROLLI_QUALITA | varchar | 255 | ✓ |
| COMMITTENZA | varchar | 255 | ✓ |
| TECNOLOGIE | varchar | 255 | ✓ |

**Indirizzo**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CAP | varchar | 50 | ✓ |
| FAX | varchar | 50 | ✓ |
| Localita | varchar | 50 | ✓ |
| POSTAELETTRONICA | varchar | 50 | ✓ |
| Provincia | varchar | 50 | ✓ |
| Regione | varchar | 50 | ✓ |
| TEL | varchar | 50 | ✓ |
| Via | varchar | 50 | ✓ |
| Codice | varchar | 50 | |

**Settore**

| Nome colonna | Tipo di dati | unghezz | Ammetti Nul |
|---|---|---|---|
| CodiceSettore | varchar | 255 | |
| DescrizioneSettore | varchar | 255 | ✓ |

**Figura 42 – Struttura db SubFor**

## 7.2.1 Sinode1

Sinode1 è composto da Fibre2Fashion, Usawear e Tessilmoda ed è stato realizzato tramite le seguenti istruzioni SQL. Ovviamente non si riportano le schermate relative al mapping dei dati in quanto già mostrate nel capitolo precedente.

```
create view     business_organization as
select          cast (categorycode as integer) as categoria, nome
from            azienda_tm
union
select          categorycode as categoria, nome
from            company_ff
```

```
create view category as
select  categorycode as codiceSettore, category as DescrizioneSettore, ' ' as SubCategory
from    settoreattivita_tm
union
select  categorycode as codiceSettore, category as DescrizioneSettore, sc.subcategory
        as SubCategory
from    category_ff as c join subcategory_ff as sc on c.subcategory=sc.subcategorycode
```

```
create view enterprise as
select  nome, descrizione, ' ' as presentazione, indirizzo, email, fax, telefono, web, url, contatti
from    company_ff
union
select  nome, descrizione, presentazione, indirizzo, ' ' as email, fax, telefono, ' ' as web, url, contatti
from    azienda_tm
union
select  nome, descrizione, ' '  as presentazione, indirizzo, email, fax, telefono, web, ' ' as url, contatti
```

```
from    company_uw


create view subcategory as
select  subcategorycode, subcategory
from    subcategory_ff
```

### 7.2.2 Sinode2

Sinode2 presenta una struttura molto simile a quella resalizzata per i file XML, per questo si riportano solo le figure relative al mapping di *goods* e *category* precedentemente omesse, e il codice relativo a tutte le viste.

```
create  view address as
select  via, cap, localita, provincia, ' ' as regione, padiglione, codice
from    azienda_im as a1 join indirizzo_im as i1 on a1.indirizzo=i1.codice
union
select  via, cap, localita,  ' ' as provincia, regione, ' ' as padiglione, codice
from    azienda_pc as a2
        join indirizzo_pc as i2 on a2.indirizzo=i2.codice
union
select  via, cap, localita, provincia, regione, ' ' as padiglione, cast (codice as integer) as codice
from    azienda_sf as a3
        join indirizzo_sf as i3 on a3.indirizzo=i3.codice


create view     business_organization2 as
select          nome, codicesettore as codiceSettore, descrizionesettore as descrizioneSettore,
                telefono, fax, email , web, ' ' as nappyweb, orari, contatti, marchicommercializzati,
                indirizzo, ' ' as referenzebancarie, ' ' as id
from            attivita_im as at
                join  azienda_im as a1 on a1.attivita=at.codicesettore
union
select          nome, codicesettore as codiceSettore, descrizionesettore as descrizioneSettore,
                telefono, fax, email, web, nappyweb, ' ' as orari, ' ' as contatti, ' ' as
                marchicommercializzati, indirizzo, ' ' as referenzebancarie, ' ' as id
from            azienda_pc as a2
                join categoria_pc as c on a2.categoria=c.codicesettore
union
select          nome, codicesettore as codiceSettore, descrizionesettore as descrizioneSettore,
                telefono, fax, email, ' ' as web,' ' as nappyweb, ' ' as orari, contatti as contatti, ' ' as
                marchicommercializzati, cast (indirizzo as integer) as indirizzo, referenzebancarie, id
from            indirizzo_sf i
                join azienda_sf as a3 on i.codice=a3.indirizzo
                join settore_sf as s on a3.settore=s.codicesettore
```

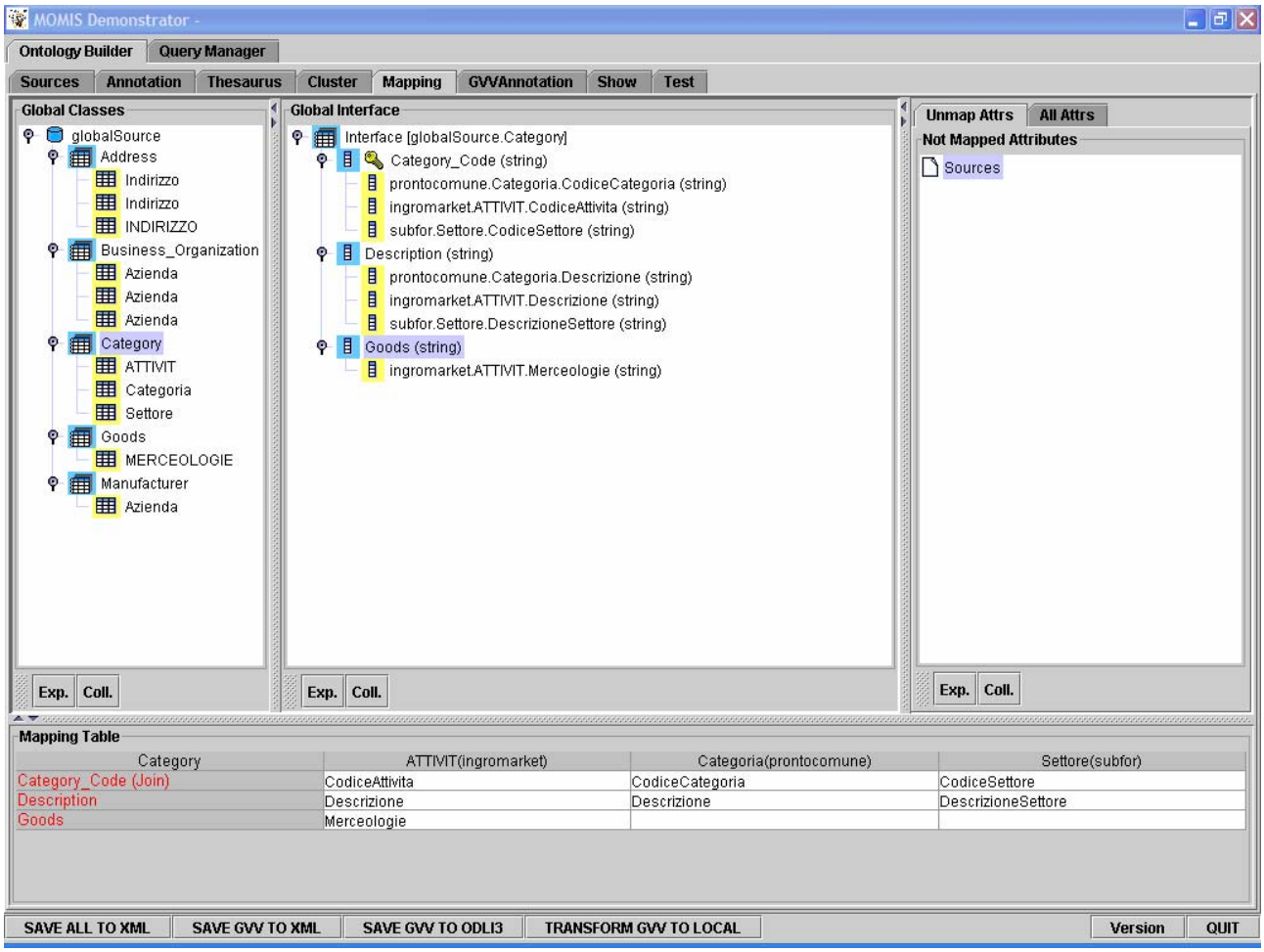



**Figura 43 – Classe globale category2**

```
create view   category2 as
select        codicesettore as CodiceSettore, descrizionesettore as DescrizioneSettore,
              merceologie as Goods
from          attivita_im
union
select        codicesettore as CodiceSettore, descrizionesettore as DescrizioneSettore, '' as Goods
from          categoria_pc
union
select        codicesettore as CodiceSettore, descrizionesettore as DescrizioneSettore, '' as Goods
from          settore_sf
```

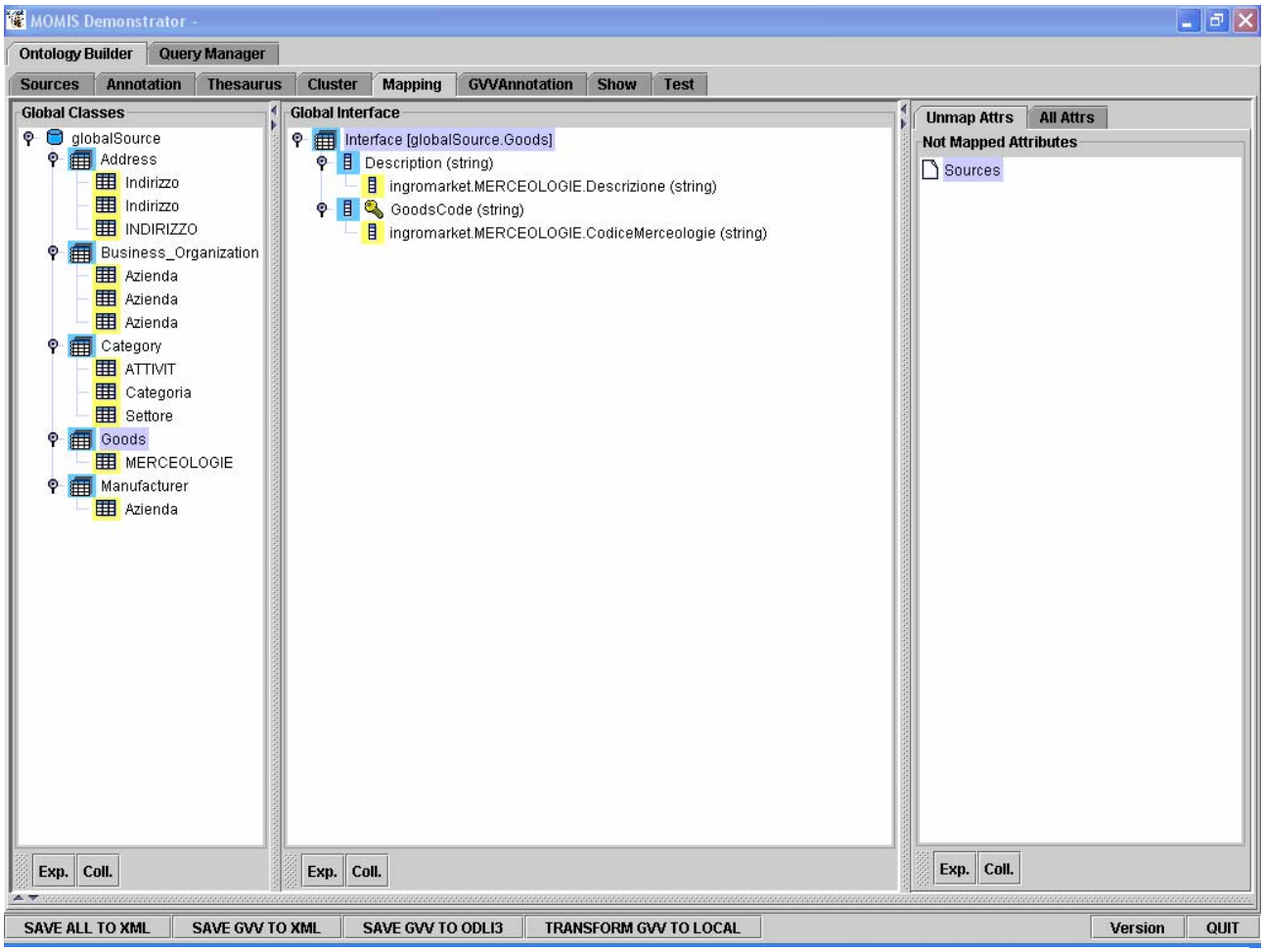



**Figura 44 – Classe globale goods**

create view **goods** as
select codicemerceologie, descrizione
from merceologie_im

create view **manufacturer** as
select nome, referenzebancarie, capitale_sociale, committenza, addetti, anno_inizio_attivita, ordini, controlli_qualita, tecnologie, turni_lavoro
from azienda_sf

## 7.2.3 Classi di secondo livello

Vengono riportati qui di seguito mapping e query inerenti al secondo livello d'integrazione.

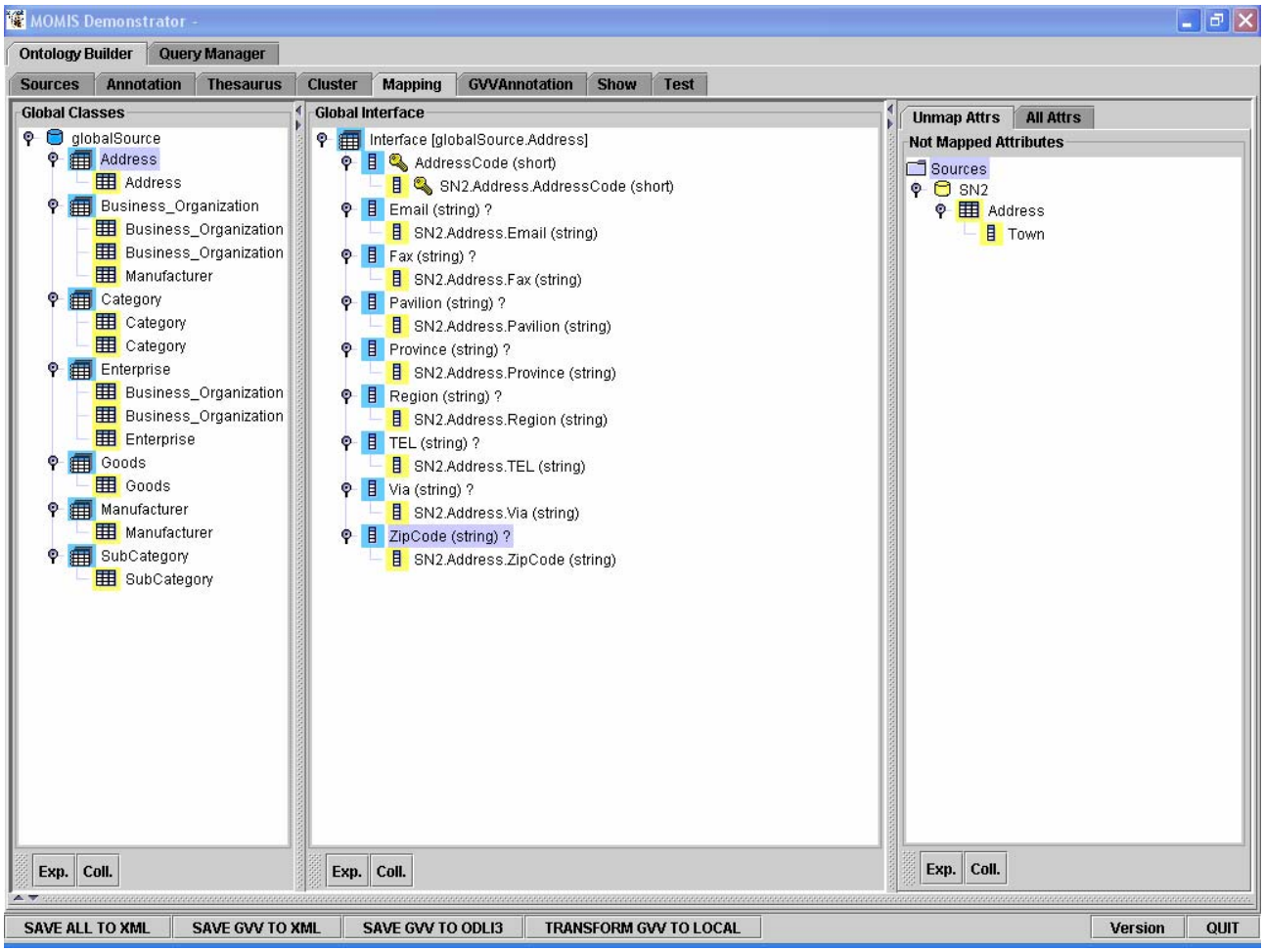



**Figura 45 – Classe globale GlobalAddress**

create view **GlobalAddress** as
select codice, via, cap, localita, provincia, regione, padiglione
from address

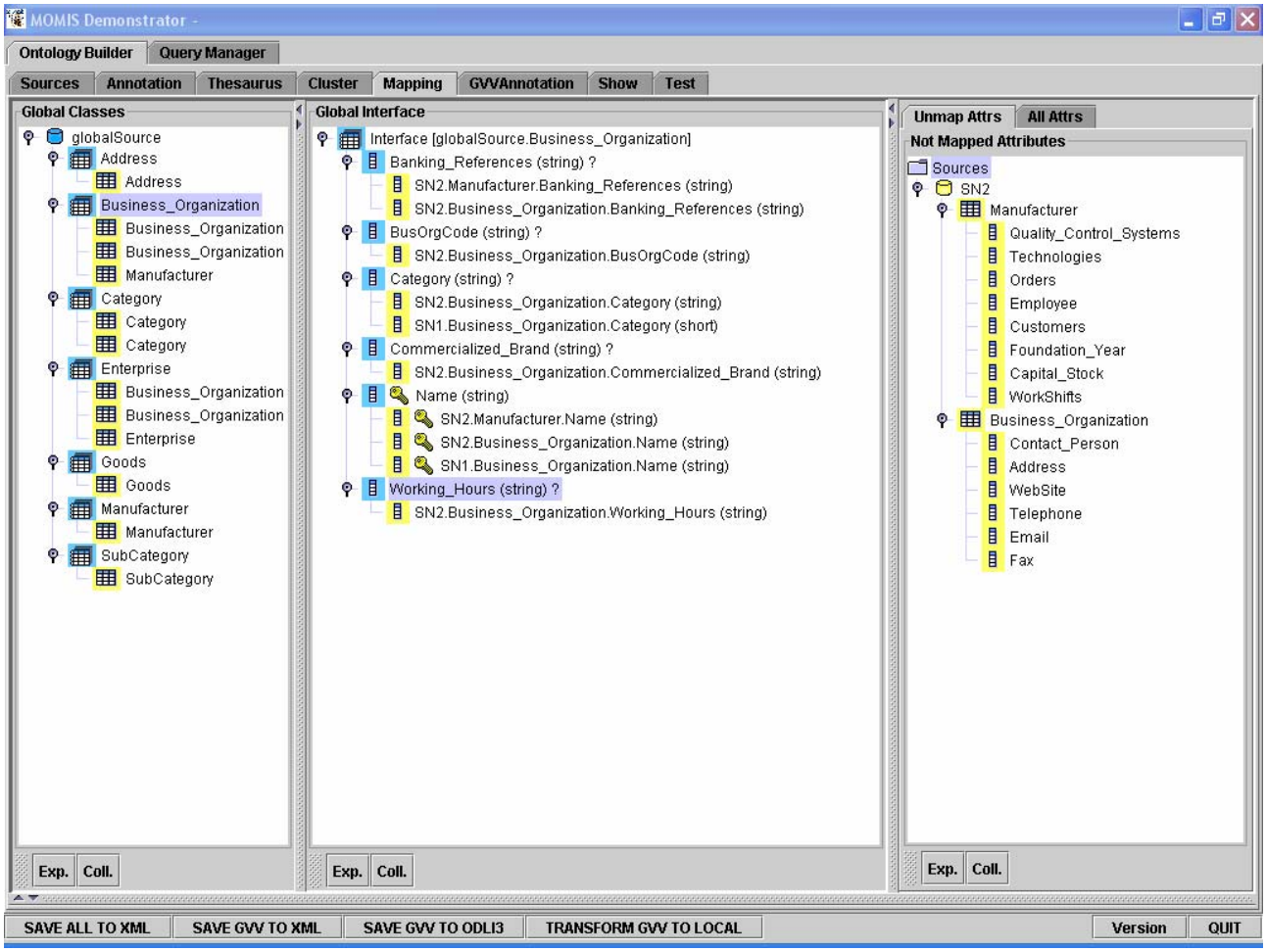



**Figura 46 – Classe globale GlobalBusiness_Organization**

create view **GlobalBusiness_Organization** as
select nome, categoria as codiceSettore, ' ' as descrizioneSettore, ' ' as marchicommercializzati, ' ' as orari, ' ' as referenzebancarie, ' ' as id
from business_organization
union
select nome, cast (codiceSettore as integer) as codiceSettore, descrizioneSettore, marchicommercializzati, orari, referenzebancarie, id
from business_organization2

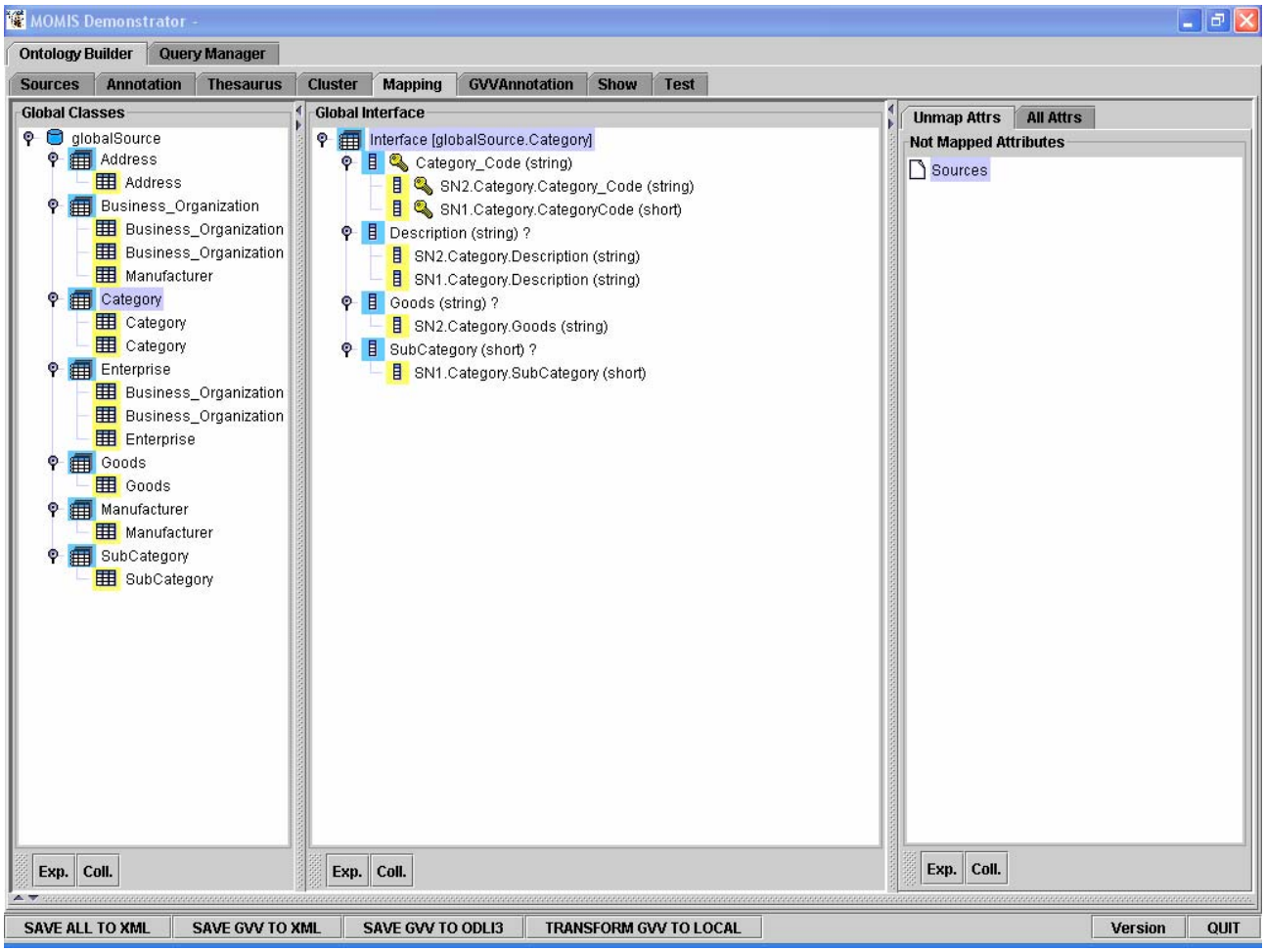



**Figura 47 – Classe globale GlobalCategory**

```
create view     GlobalCategory as
select   codiceSettore, cast(descrizioneSettore as varchar(100)) as descrizioneSettore, ' ' as goods,
         subcategory
from     category
union
select   cast (codiceSettore as integer) as codiceSettore, descrizioneSettore, goods, ' ' as subcategory
from     category2
```

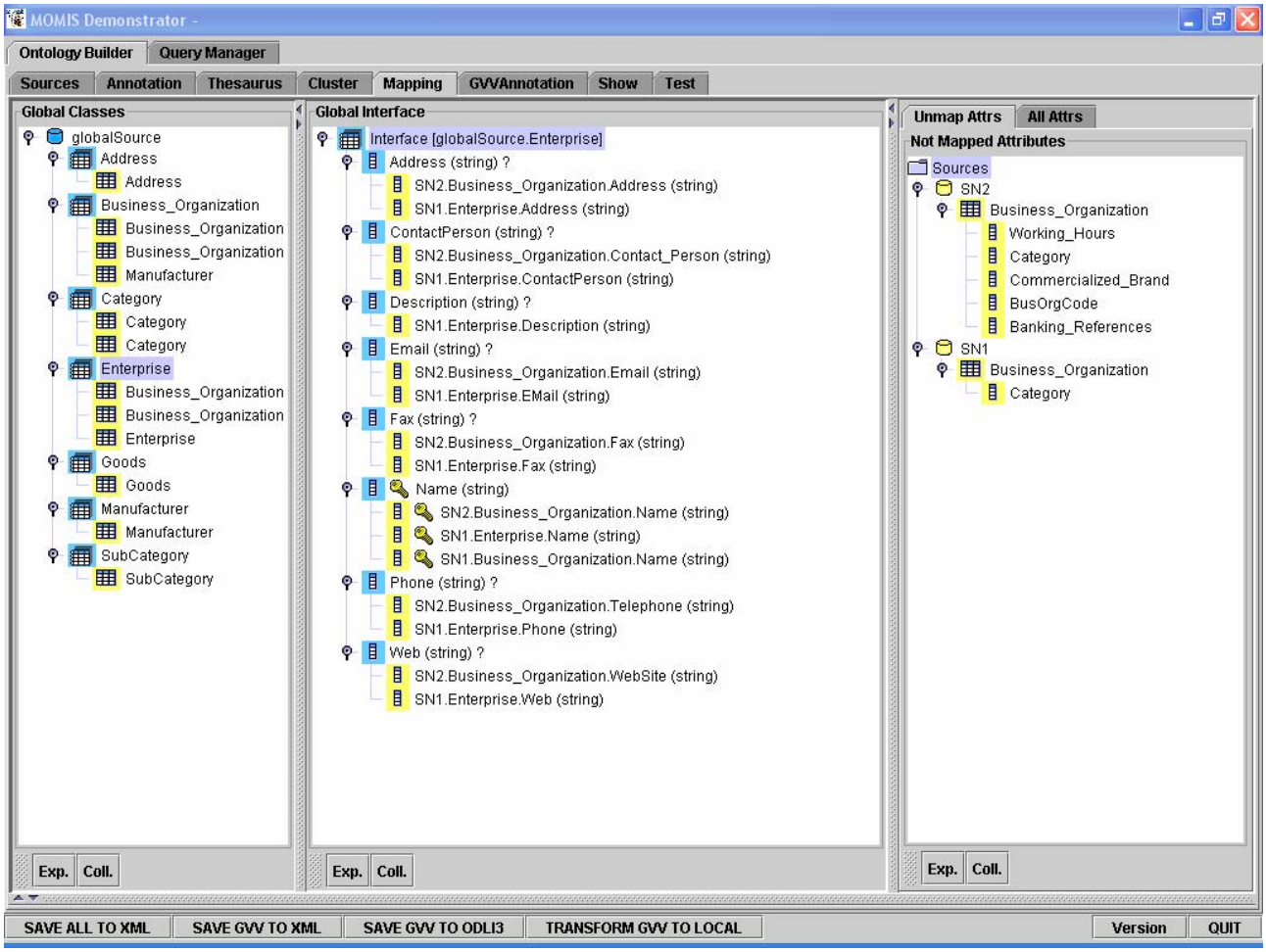



**Figura 48 – Classe globale GlobalEnterprise**

create view **GlobalEnterprise** as
select nome, via, cap, localita, provincia, regione, padiglione, telefono, fax, email, web, nappyweb,
' ' as url, ' ' as descrizione, contatti
from business_organization2 as b_o
join Globaladdress as g_a on b_o.indirizzo=g_a.codice
union
select nome, indirizzo as via, ' ' as cap, ' ' as localita, ' ' as provincia, ' ' as regione, ' ' as padiglione,
telefono, fax, email, web, ' ' as nappyweb, url, descrizione, contatti
from enterprise

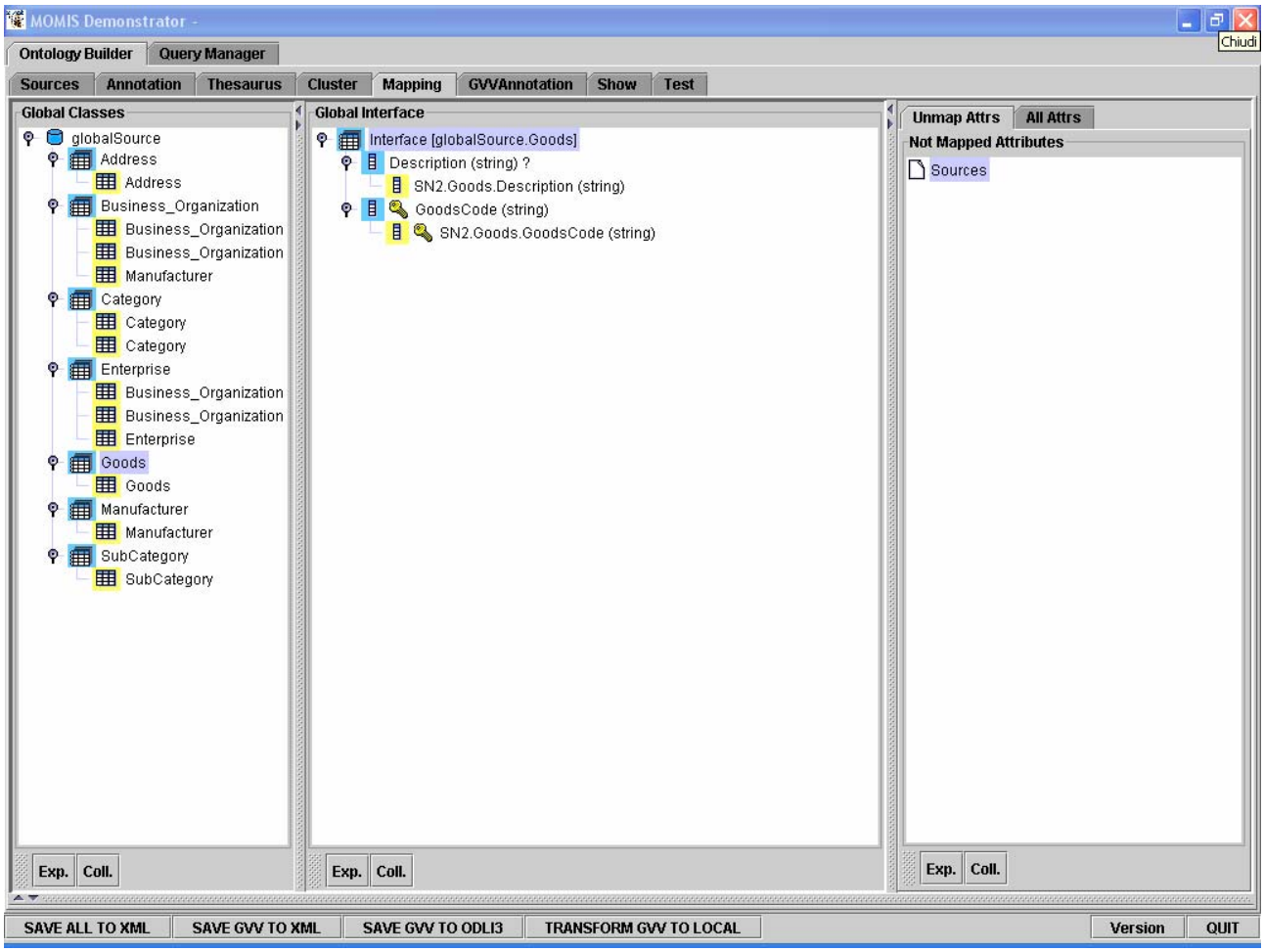



**Figura 49 – Classe globale GlobalGoods**

create view **GlobalGoods** as
select *
from goods

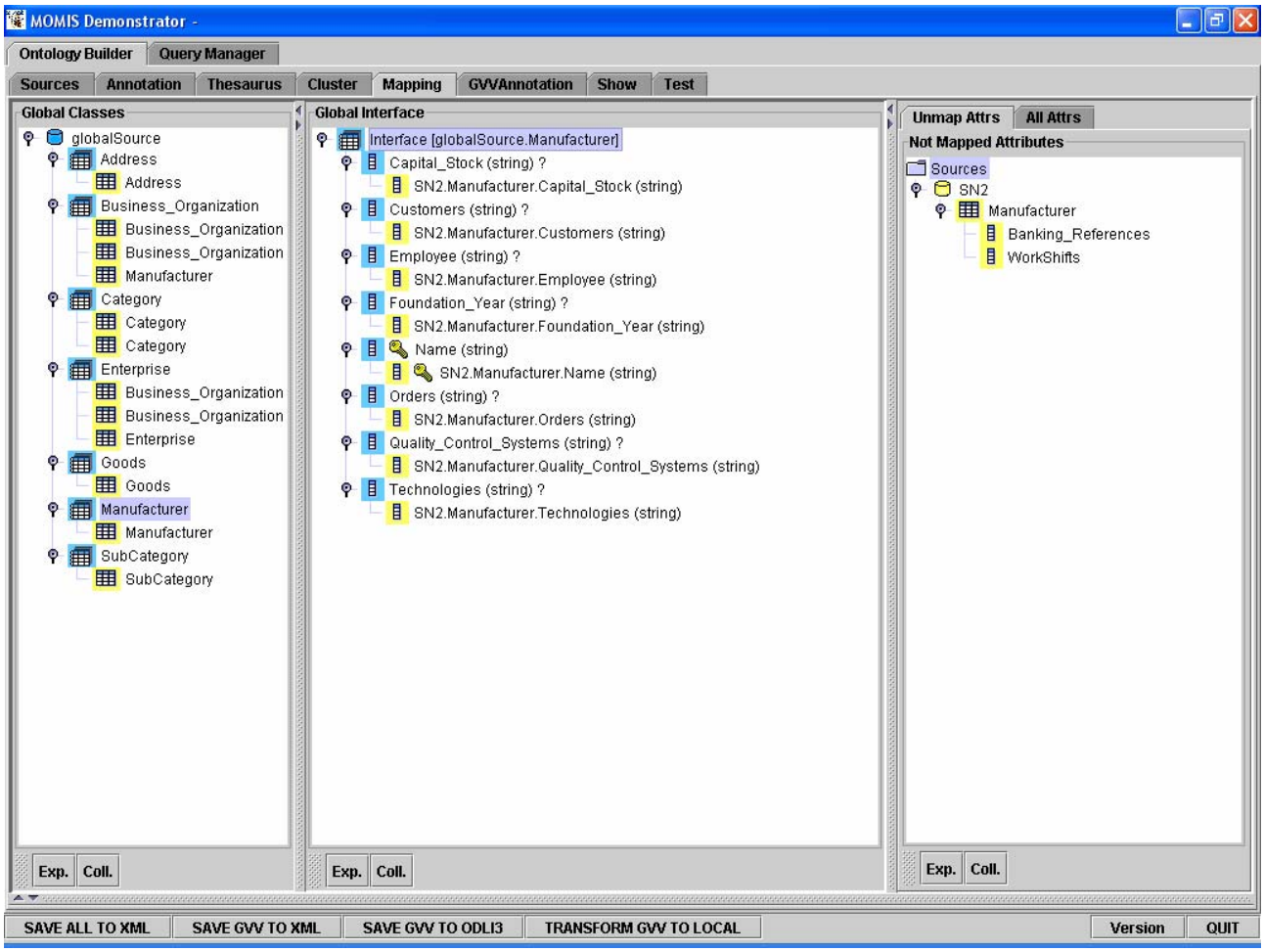



**Figura 50 – Classe globale GlobalManufacturer**

create view **GlobalManufacturer** as
select nome, capitale_sociale, committenza, addetti, anno_inizio_attivita, ordini, controlli_qualita, tecnologie
from manufacturer

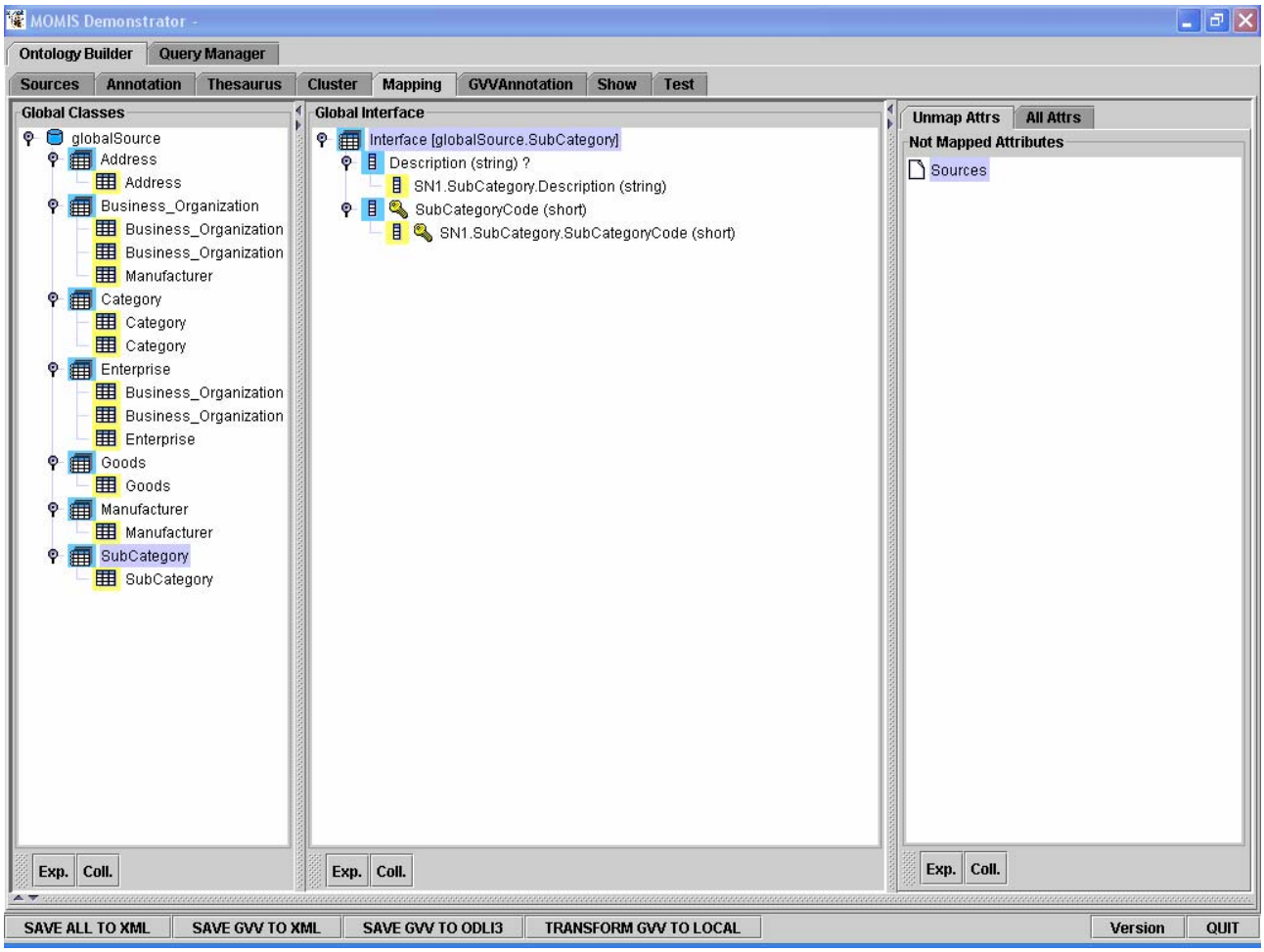



**Figura 51 – Classe globale GlobalSubCategory**

create view **GlobalSubCategory** as
select *
from subcategory

## 7.2.4 Test dell'ambiente creato

Per appurare che la struttura realizzata fosse corretta e per accertare che le funzioni fornite dal linguaggio SQL siano utilizzabili anche in contesti di questo tipo, si sono sottoposte le viste di secondo livello ad una serie di interrogazioni atte, appunto, a verificare i suddetti requisiti. Si è constatato che tutte le considerazioni effettuate nel paragrafo 7.1.2 sono ancora valide; le query eseguite e i risultati ottenuti non vengono riportati perché molto simili a quelli già presentati in precedenza. A conferma di quanto affermato, comunque, viene riportato un esempio che seleziona il nome, la descrizione, il numero telefonico e i marchi commercializzati di tutte le aziende il cui settore è *importer*.

Select distinct e.nome, e.descrizione, e.telefono, b_o.marchicommercializzati
from GlobalBusiness_Organization as b_o, GlobalCategory as c, GlobalEnterprise as e
where e.Nome=b_o.Nome
and c.DescrizioneSettore='Importer'

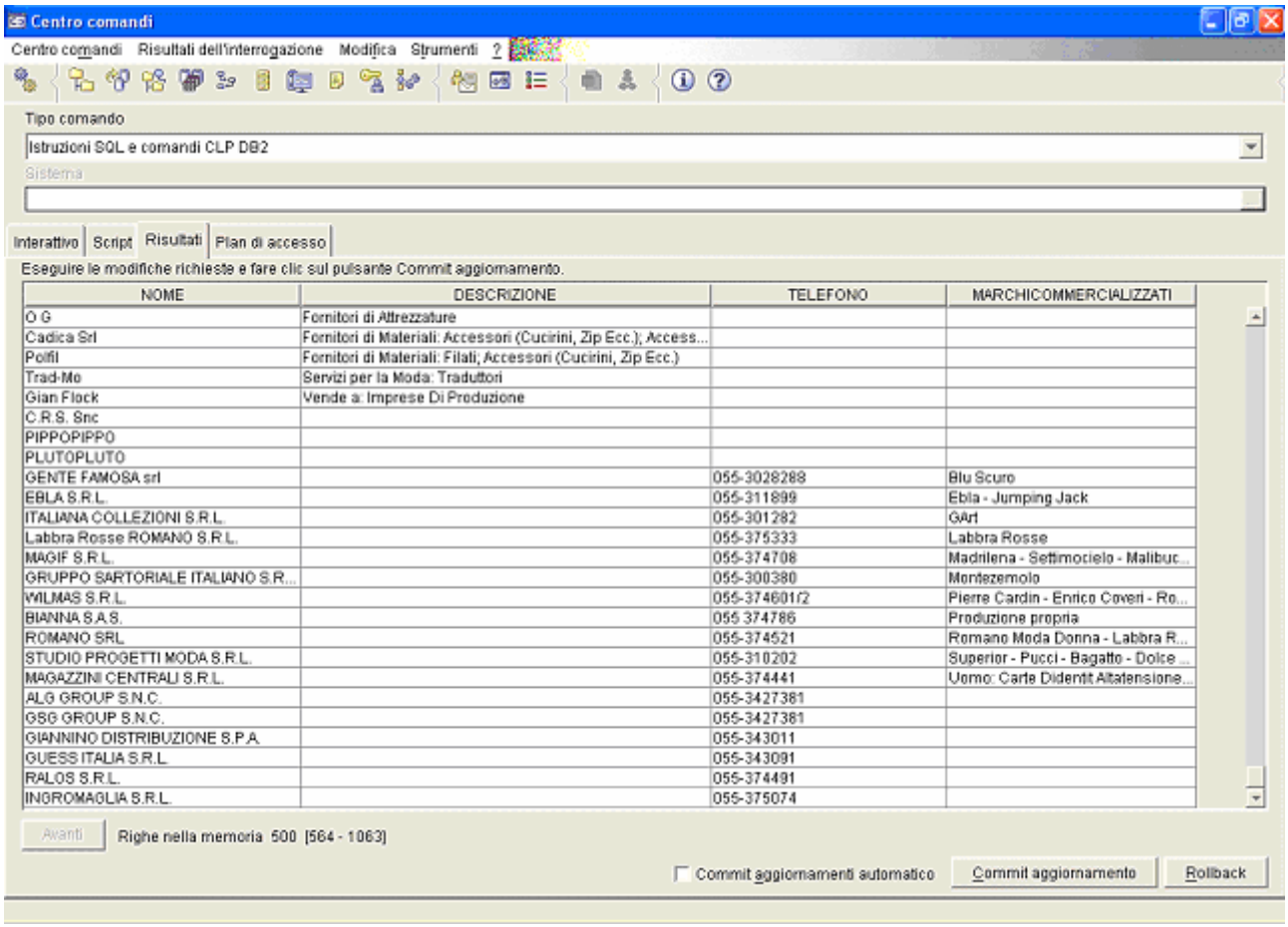



**Figura 52 – Risultati query**

# Conclusioni

L'analisi portata avanti sin ora ha messo in luce i vari aspetti dei due tool ma non ha ancora effettuato un confronto vero e proprio se non in modo disparato e disaggregato. Questo capitolo conclusivo si propone, appunto, come sunto di quanto detto ed esaminato prefiggendosi l'obiettivo primario di chiarire meglio le varie lacune e i vari punti di forza dei due software.

Le potenzialità di DB2 Information Integrator sono già state ampliamente esposte nei capitoli precedenti e, da quanto affermato, si capisce che esso rappresenta la nuova generazione di middleware d'integrazione dell'informazione; in un'unica soluzione permette la copertura di applicazioni che vanno dai tool analitici fino ai portali consentendo un accesso trasparente e distribuito a dati risiedenti in differenti locazioni. Il fatto che sia fondato su standard industriali come SQL, Java e XML rende possibile l'ottimizzazione degli investimenti tecnologici e delle strutture dati già esistenti all'interno di un'azienda.
In un articolo dello scorso anno del CRN gli analisti del settore industriale lodarono le recenti proposte dell'IBM dichiarando che il punto di forza della compagnia è rappresentato dalla sua intrinseca logica di ottimizzazione che non è seconda a nessuno e dalla capacità di padroneggiare la gestione di dati sia strutturati che non con grande semplicità. Inoltre si legge che la società possiede le potenzialità per sormontare l'ostacolo dell'integrazione dell'informazione all'interno delle imprese avendo molto da offrire rispetto a compagnie rivali; infatti, mentre la concorrenza riesce a coprire solo una ristretta fetta del mercato dell'integrazione, l'IBM può elargire una vasta, dettagliata e completa piattaforma d'integrazione attraverso prodotti che cooperano ininterrottamente. Tutto questo grazie agli oltre trenta anni di esperienza nella realizzazione e nello sviluppo di strumenti base per il middleware che permettono a tecnologie differenti di cooperare in svariati ambienti diversi tra loro.[4]

D'altra parte, però, anche il sistema MOMIS può essere considerato come un valido strumento per l'integrazione semi-automatica di schemi eterogenei, tramite la realizzazione di una GVV permette, come già osservato, di porre interrogazioni senza preoccuparsi dell'implementazione delle sorgenti sottostanti. Esso è ancora in fase sperimentale ovvero si presenta in una forma, comunque funzionante, ma non ancora del tutto completa che potrà essere ottimizzata e perfezionata nel tempo.

## 8.1 Software di appoggio

Mentre DB2 Information Integrator richiede solamente il DBMS IBM DB2, per utilizzare MOMIS sono necessari sia MySQL che SQL Server con tutte le ben note problematiche relative agli investimenti e ai costi che un'azienda dovrà sostenere per ottenere una piattaforma compatibile col tool.

## 8.2 Importazione delle sorgenti

La facilità del procedimento d'importazione delle sorgenti risulta essere indicativamente la stessa in entrambi i software, se però da una parte, DB2 Information Integrator permette di integrare qualsiasi tipo di sorgente, MOMIS comprende per ora solo un ristretto insieme di Wrapper che offrono la possibilità di importare solamente database Access, Oracle, SQL Server, file XML, e sorgenti in formato $ODL_I$[3] e OWL (linguaggio della W3C per le ontologie di XML). È importante comunque sottolineare che è possibile espandere questo insieme attraverso la definizione di nuovi moduli per l'estrazione delle strutture degli schemi appartenenti a sorgenti non ancora previste.

[4] "IBM: Xperanto Rollout to Start In Early 2003", http://www.crn.com/sections/BreakingNews/breakingnews.asp?ArticleID=39187

## 8.3 Il mapping

A mio parere il processo di mapping rappresenta l'aspetto più interessante di MOMIS. Si potrebbe pensare che la fase di annotazione rallenti di molto il processo d'integrazione specialmente se si stanno trattando quantità abbastanza elevate di informazioni, ma è necessario non dimenticare due caratteristiche fondamentali di questa funzionalità. La prima è che essa può essere tralasciata comportando che il mapping debba essere realizzato manualmente, procedimento che comunque risulta piuttosto rapido e semplice attraverso l'interfaccia grafica. La seconda è che è molto probabile che siano presenti ripetizioni di attributi con gli stessi significati all'interno delle varie entità e ciò, per quanto detto nel paragrafo 6.2.2, accelera notevolmente questa fase.

Indubbiamente il difetto principale di DB2 Information Integrator è quello che obbliga il progettista a conoscere gli schemi delle sorgenti per poter porre le interrogazioni sui nickname nonché la necessità di dover indicare esplicitamente i campi e le condizioni di join per query che coinvolgono più tabelle. Come detto precedentemente l'integrazione degli schemi non avviene in modo semi-automatico, è necessario alterare i tipi di dato dei campi dei nickname per poter realizzare le classi globali attraverso viste; va comunque sottolineato che l'IBM presso l'Almaden Research Laboratory e l'Università di Toronto sta portando avanti un progetto, denominato CLIO, il cui obiettivo primario è quello di facilitare l'utente nella fase d'integrazione di schemi e nella scrittura d'interrogazioni che coinvolgono più schemi. Infine, questo tool, non affronta la gestione dei cosiddetti dati sporchi, dirty data, i quali si presentano in situazioni che prevedono, ad esempio, uno schema $\alpha$ nel quale esiste una tabella con il campo *customer number* il cui significato è il numero del cliente e uno schema $\beta$ nel quale è presente un'altra tabella con lo stesso attributo che, però, indica il numero di tessera sanitaria. Questo problema non viene, appunto, preso in considerazione da DB2 Information Integrator che delega la sua risoluzione a strumenti esterni al pacchetto come, ad esempio, tool ETL.

## 8.4 Aggiornamento dei nickname

DB2 Information Integrator generalmente permette anche funzioni di scrittura oltre che a quelle di lettura finora esaminate. Questo rende possibile aggiungere record contemporaneamente sia ai nickname che alle tabelle remote tramite un semplice INSERT INTO. Se poi è stato realizzato un mapping come quello creato per l'esempio esposto si osserverà che, ovviamente, le viste che rappresentano le classi globali vengono automaticamente aggiornate dal sistema e, una volta richiamate, presentano le nuove informazioni inserite.
Questa potente funzionalità però non può essere utilizzata in casi che riguardano file XML per i quali non è possibile aggiungere dati agli oggetti importati.

L'inserimento di nuove tuple è anche consentito in strutture come le viste purché la loro definizione lo permetta; in particolare è possibile aggiungervi record se esse rappresentano semplici proiezioni o selezioni di tabelle in cui non sono definite condizioni di join. Per testare questa funzionalità si è provato, sempre tramite il comando INSERT INTO, ad aggiungere un record alla vista *GlobalGoods* e si è osservato che le nuove informazioni erano presenti sia nella vista stessa, sia nella vista *Goods* dalla quale deriva, nel nickname corrispondente *merceologie_im* ed, infine, anche nella tabella *merciologie* del database remoto di SQL Server.

Un altro aspetto che si è analizzato, inerente all'aggiornamento dei nickname, è quello riguardante la creazione e la cancellazione di colonne eseguita sia in ambiente locale che remoto. Mentre non è possibile portare a termine nessuna di queste due operazioni sui nickname, si assiste ad un comportamento del tool differente a seconda dell'intervento da effettuare quando queste operazioni vengono eseguite sulle sorgenti e che, in entrambi i casi, risulta essere comunque non del tutto positivo. Se si cancella una colonna di una tabella remota ogni volta che si effettueranno operazioni

inerenti ad essa, DB2 Information Integrator invierà un messaggio di errore all'utente in quanto una colonna appartenente alla tabella locale ne referenzia una non più esistente. Se, al contrario, viene aggiunta una nuova colonna ad una tabella remota tale attributo non comparirà nella struttura del nickname il quale però rimane, come nel caso precedente, utilizzabile anche se non aggiornato. Per una piena simmetria e funzionalità si consiglia comunque di reimportare i nickname ogni volta che le tabelle remote subiscono modifiche di struttura; non è, invece, necessario eseguire questa operazione se tali entità acquisiscono nuovi record in quanto, si ricorda, il tool opera on the fly e quindi permette sempre di avere dati aggiornati in tempo reale.

Si è inoltre verificato che è possibile modificare i valori di determinati campi in modo tale che questi aggiornamenti abbiano effetto sia in locale che in remoto così come nell'esempio riportato:

```
update azienda_im
set    telefono='0522/302291'
where  nome=' EVASION BY LA FULARISSIMA S.A.S'
```

Per quanto riguarda l'ambiente MOMIS la situazione non si presenta molto diversa. Infatti anche in esso non è possibile, per il momento, né aggiungere né eliminare colonne dalle tabelle locali, e le modifiche relative all'eliminazione o all'inserimento di attributi di tabelle remote producono errori quando una query va ad interrogare quei determinati campi proprio come accade con DB2 Information Integrator. Il discorso qui però assume dimensioni più grandi in quanto viene negata, all'utente, la possibilità di apportare qualsiasi cambiamento alle sorgenti importate non permettendo dunque neppure l'inserimento, la cancellazione o la modifica di un record; è stato comunque verificato che rieseguendo una query su di una tabella le cui tuple hanno subito modifiche a livello remoto si ha la possibilità di visualizzare i dati aggiornati.

Quanto affermato ritrova un riscontro in due approcci tra loro differenti che sono, però, caratterizzanti per i due software. Mentre DB2 Information Integrator opera, come già detto, seguendo il metodo federato che permette, grazie a DB2, l'aggiornamento dei record, MOMIS appartiene alla classe dei mediatori ovvero tool che realizzano viste virtuali appositamente in sola lettura al fine di garantire l'integrità delle sorgenti con cui interagiscono.

## 8.5 Set di istruzioni SQL

Come si è già visto il set d'istruzioni SQL offerto da MOMIS è molto esiguo soprattutto se paragonato a quello fornito da DB2 Information Integrator che risulta essere quasi completo. Se da una parte è bene rimarcare che anche quest'aspetto potrà, in futuro, essere perfezionato dai progettisti di MOMIS, dall'altra occorre sottolineare che per ora tali istruzioni possono essere eseguite solo su sorgenti relazionali e non, ad esempio, su sorgenti XML il cui Wrapper non è ancora in grado di tradurre le richieste SQL in XQuery.

## 8.6 Esportazione degli ambienti creati

Mentre MOMIS permette di salvare gli scenari creati in file XML o OWL in modo da poterli esportare e riutilizzare su altre macchine, DB2 Information Integrator ricorre ad un approccio sicuramente più tradizionale rappresentato dalla realizzazione del backup del database nel quale si è realizzato il processo d'integrazione. Risulta ovvio che una volta che il database sarà ripristinato conterrà tutti gli oggetti precedentemente creati o importati.

## 8.7 Quale tool scegliere?

In base allo studio eseguito e alle informazioni riportate, effettuare una scelta fra i due software non risulta semplice. Ovviamente se si considera MOMIS nella sua forma attuale risulta ovvia una propensione verso il tool IBM; ma se si prescinde da ciò e si considera il primo in una forma più completa ed ottimizzata come quella a cui sicuramente approderà in futuro la scelta non risulta essere così immediata e banale.

Non ritenego adeguato fare un paragone di questo tipo tra un prodotto ancora in fase di sviluppo e uno già completamente definito ma è chiaro che se da una parte si assume che le funzionalità di MOMIS potranno essere ampliate, dall'altra non è possibile ignorare il fatto che anche DB2 Information Integrator potrà, fra qualche anno, essere perfezionato e migliorato.

Allo stato attuale sicuramente situazioni che richiedano l'integrazione di sorgenti allo scopo di realizzare classi globali nelle quali vengono mappati i vari attributi, vedono in MOMIS la soluzione più efficace sempre che le interrogazioni che verranno effettuate risultino essere semplici e le tipologie di sorgenti interessate rientrino in quelle previste. Se, al contrario, si devono trattare altri tipi di sorgenti o query piuttosto complesse occorre necessariamente ricorrere a DB2 Information Integrator richiedendo un maggior sforzo da parte del progettista che si vedrà costretto a realizzare viste, in alcuni casi anche molto articolate, con le problematiche esposte precedentemente. Altra circostanza in cui è indispensabile l'utilizzo del tool IBM è quella che prevede la possibilità di inserire o cancellare nuovi record direttamente nei nickname e fare in modo che queste modifiche vengano apportate anche nella sorgente corrispondente.

Ci tengo, infine, ad esprimere il mio personale apprezzamento per MOMIS in quanto ritengo che abbia grosse potenzialità e che potrà assumere, una volta completato, un ruolo importante nell'ambito dell'integrazione dell'informazione.

# Indice delle figure

# Bibliografia


***White papers*** http://www-306.ibm.com/software/data/pubs/papers/#iipapers

- **Information Integration – distribuited acces and data consolidation**, Berry Devlin, marzo 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/etl.pdf
- **Creating a flexible infrastructure for information integrator**, Holly A. Hayes, ottobre 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/ii.pdf
- **Using the federated database technology of IBM DB2 Information Integrator**, Anjali Grover, Eileen Lin, Ioanda Ursu, ottobre 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/iifed.pdf
- **Enterpise data access with DB2 Information Integraator for Content**, Carol S. Greenstreet, marzo 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/iiforcontent.pdf
- **Inforantion Integration – Extending the data warehouse,** Berry Devlin, marzo 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/exendingdw.pdf
- **The real Data Warehousing Story with DB2 Universal Database and DB2 Information Integrator,** IBM Software Group Toronto Lab, ottobre 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/realdw.pdf
- **DB2 information Integrator XML Wrapper Performance**, IBM Software Group Sylicon Valley Lab, ottobre 2003
  ftp://ftp.software.ibm.com/software/data/pubs/papers/db2iixmlwrapper.pdf

- **IBM DB2 Information Integration home page** www.ibm.com/software/data/integration

- **IBM DB2 Information Integrator website** www.ibm.com/software/data/integration/db2ii

- **01Net - LineaEDP** http://www.01net.it/01NET/HP/0,1254,1_ART_40837,00.html

- Holly A. Hayes, "**DB2 Information Integrator: The Big Picture**" Luglio 2003
  http://www-106.ibm.com/developerworks/db2/library/techarticle/0309baeurle/0309baeurle.html

- Holly A. Hayes, Nelson Mattos, "**Information On Demand**",
  http://www.db2mag.com/story/ahowArticle.jhtml?articleID=12803103

- Aberdeen Group, Inc. "**IBM DB2 Information Integrator: Scope, Power, Service**", 2003
  http://www-306.ibm.com/software/data/services/ii.html
  ftp://ftp.software.ibm.com/software/data/pubs/tech-consult/eii.pdf

- "**IBM DB2 Information Integrator: integrated, on demand access to distributed information**", Settembre 2003
  http://www-306.ibm.com/software/data/info/literature/brochures.jsp
  ftp://ftp.software.ibm.com/software/data/pubs/brochures/iiexecbrief.pdf

- "**IBM DB2 Information Integrator, Version 8.1**"
  http://www-306.ibm.com/software/data/info/literature/brochures.jsp
  ftp://ftp.software.ibm.com/software/data/pubs/brochures/iispecsheet.pdf

- "**IBM DB2 Information Integrator Services Reference Catalog**", Aprile 2004
  http://www-306.ibm.com/software/data/services/ii.html
  ftp://ftp.software.ibm.com/software/data/services/iiservices.pdf

- "**IBM Design and Planning Service for DB2 Information Integrator**"
  http://www-306.ibm.com/software/data/services/ii.html
  ftp://ftp.software.ibm.com/software/data/services/design-planning-db2v13.pdf

- "**IBM Implementation Service for DB2Informatoin Integrator**"
  http://www-306.ibm.com/software/data/services/ii.html
  ftp://ftp.software.ibm.com/software/data/services/implementationservices_ii.pdf

- Felignani Alain "**IBM DB2 Information Integrator**", appunti

- Felignani Alain "**DB2 Information Integrator e MOMIS**", appunti

- Università degli studi di Modena e Reggio Emilia, **Momis home page**
  www.dbgroup.unimo.it/Momis

- D. Beneventano, S. Bergamschi, F. Guerra, M. Vincini, "**The MOMIS approach to information integration**", IEEE e AAAI International Conference on Enterprise Infromation System, Luglio 2001 http://dbroup.unimo.it/prototipo/paper/iceis01.pdf

***Tutorial*** http://www-106.ibm.com/developerworks/db2/library

- **Configuring IBM DB2 Information Integrator to Access Diverse Data**, Larissa N. Wojciechowski
- **Developing a VB.NET Federated Application for Microsoft Access**, Abdul Al-Azzawe
- **Asinchronous information integration**, Jacques Labrie
- **Access heterogeneus data with DB2**, Julien Muller