Seminari di Ingegneria del software

Introduzione alla Data Integration

e

Integrazione dei dati dei Sistemi Informativi

di un

ente di formazione

Di

Riccardo Cocetta

r.cocetta@dreamarts.it

Stampato mercoledì 14 gennaio 2009

Indice

Obiettivo

Lo scopo della presente tesina è quello di presentare i principi di base della Data Integration e come esempio fornire un modello per l'integrazione dei dati presenti negli archivi di un ente di formazione.

Motivazioni

L'ente di formazione oggetto di questo lavoro utilizza diversi sistemi interni sui quali è presente parte dell'informazione relativa ai propri allievi, ed alcuni portali esterni nei quali sono presenti altre informazioni; questo comporta una grave perdita di tempo ogni qualvolta si venga a creare la necessità di recuperare i dati relativi agli account degli allievi sui diversi sistemi o di risalire da un account ad un contatto effettuato dal reparto marketing.

Lo stesso discorso vale per gli operatori, i loro account e permessi sui vari sistemi, che dipendono dal reparto di appartenenza, e per le informazioni relative ai vari progetti formativi, e gli enti che bandiscono i finanziamenti per i corsi.

Si vuole quindi con questa tesina, proporre la Data Integration come metodo per porre rimedio a questi problemi, e fornire un modello teorico che possa essere utilizzato in futuro come base per un eventuale progetto di integrazione.

Verranno quindi prima presentati i principi della Data Integration per poi andare a modellare un caso reale.

Introduzione alla Data Integration

Cosa è e a cosa serve

Nella organizzazione delle aziende, il valore dell'informazione diventa sempre più elevato, e la capacità da parte delle stesse di poterle utilizzare per monitorare i processi, misurarne le prestazioni, guidare i flussi operativi e supportare le decisioni direzionali decreta la loro possibilità di essere competitive nei mercati in cui operano.
Negli ultimi anni si è assistito ad un aumento dei sistemi informatici utilizzati per gestire tutti i processi aziendali che vanno dal marketing alla produzione, dall'amministrazione alla gestione del post-vendita, fino al monitoraggio sull'erogazione di servizi, e troppo spesso questi sistemi sono stati introdotti senza considerare l'interoperabilità delle funzioni aziendali, l'importanza della trasferibilità delle informazioni da un processo all'altro ed i problemi generati dall'overload informativo.
Il proliferare dei sistemi informatici, e l'ampliamento dello spettro delle informazioni trattate nelle basi di dati dei sistemi informativi aziendali, portano alla disgregazione dei dati in diversi frammenti informativi che rendono difficile l'uso degli stessi per monitorare i processi aziendali e dare supporto alle decisioni. È necessario quindi proporre dei metodi e dei modelli che consentano l'integrazione dei dati sparsi nei vari sistemi informativi, e li rendano raggiungibili ed utilizzabili come se fossero appartenenti ad un'unica base di dati, permettendo quindi alle aziende di rivalutarne il contenuto informativo perso nella frammentazione.
Per Data Integration si intende il processo di integrazione di dati provenienti da diverse ed eterogenee fonti informative, dove per fonti eterogenee si intende ad esempio il fatto che queste possono essere costituite da diversi tipi di basi di dati, file di testo, file XML, altri tipi di file, o ad esempio possono essere l'output di un programma. In sostanza con la Data Integration si vuole fornire un unico punto dove effettuare query su un insieme di sorgenti di dati tra loro diverse.

Avere una visione globale di tutti i dati riguardanti l'azienda e delle connessioni fra le diverse fonti è il primo passo necessario per implementare soluzioni strategiche di Business Intelligence , di Competitive Intelligence, e di Knowledge Management.

Come funziona

Modello base

Alla base della Data Integration c'è l'idea di permettere la risposta ad una query eseguita sulle diverse fonti aziendali, come se fosse fatta su un'unica base di dati (fig.1). Per poter permettere questo si vogliono rappresentare tutte le fonti secondo un unico Global Schema, che raccolga a livello concettuale le relazioni esistenti in una fonte, e le integri con le informazioni presenti nelle altre.

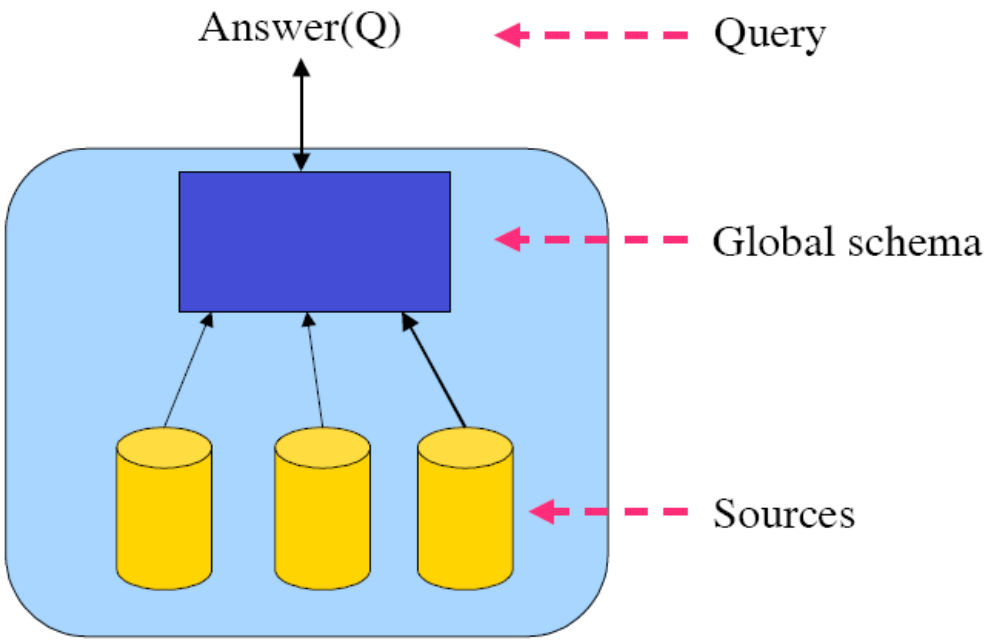

Figura 1 - Modello base della Data Integration

Le sorgenti

All'interno dei sistemi informativi si possono individuare diverse collezioni di informazione organizzate in database, singole tabelle, file xml, file di testo, od anche informazioni prodotte da una singola funzione di un programma come output.

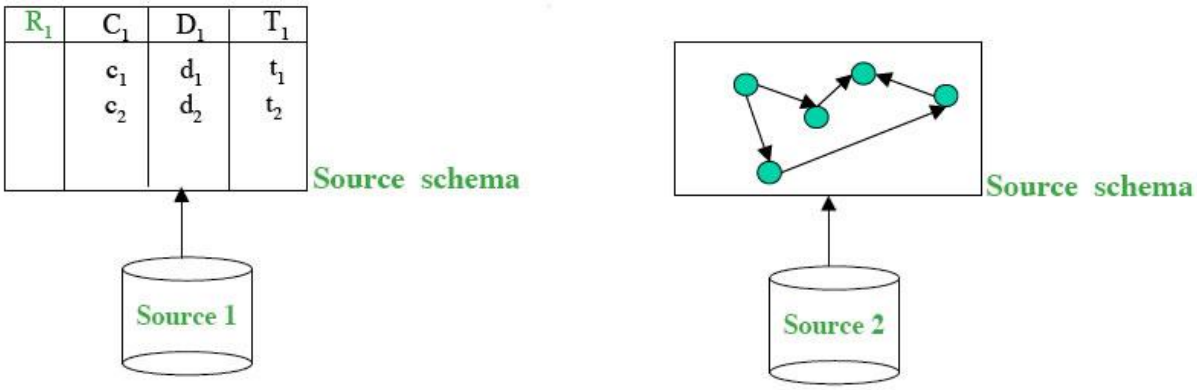

Figura 2 – Sorgenti

Queste collezioni sono individuate come sorgenti, o fonti e sono l'oggetto di partenza del lavoro della data integration, in quanto contengono i dati che siamo interessati a integrare in un uno schema globale.

Un esempio di sorgente potrebbe essere l'archivio clienti del software gestionale (eventualmente con un database di Microsoft Access) di una azienda, un altro la tabella degli account utente sul sito web (per esempio in MySQL), la tabella contenente dei feedback sui prodotti, e le schede prodotto presenti sul sito in formato XML, e ad esempio potrebbe esserci un sistema ERP dedicato alla produzione nel quale vengono anche registrati i difetti dei prodotti per poterne misurare la qualità. L'azienda potrebbe voler correlare la qualità in produzione con i difetti di un determinato componente di un prodotto segnalati dagli utenti sul sito, e nella situazione descritta questo sarebbe impossibile se non con una correlazione manuale.

Un secondo esempio potrebbe essere dato da una azienda che abbia un sistema di mail marketing, e voglia misurare l'efficacia del sistema stesso, ovvero capire quale sia la correlazione tra le e-mail inviate ed il cambio di fatturato. Se si considerano due sistemi divisi (come spesso accade), un gestionale per il fatturato ed un sistema di invio mail per il mail marketing, ottenere questa informazione in modo automatico diventa veramente difficile se non impossibile.

Le fonti di cui abbiamo parlato sinora sono fonti dette **intra-organizzazione**, ovvero che sono presenti solamente all'interno di una organizzazione, ma è possibile dover utilizzare fonti **inter-organizzazione** ovvero dati provenienti da più organizzazioni.
Un esempio di fonti inter-organizzazione può essere dato ad esempio da un sito che si occupa di vendita di componenti Hardware (un mercato estremamente veloce) e che per dare una migliore informazione ai propri clienti prenda informazioni relative ai prodotti e alle loro caratteristiche tecniche direttamente dai database dei suoi fornitori. In questo caso risulta evidente la possibilità che le fonti da cui si parla siano eterogenee, e quindi che siano gestite con formati diversi e abbiano anche una struttura diversa, perché mentre un fornitore potrebbe dare accesso ad un suo Web Service per ottenere il listino e le schede tecniche, altri potrebbero fornire un file CSV separato da virgole per dare le stesse informazioni, e magari qualcuna in più che non serve allo scopo del sito. Lo scopo base della Data Integration è appunto dare un accesso trasparente a queste fonti, come se facessero parte di un unico database.

Il global schema

Il global schema rappresenta la visione che si vuole dare all'utente dei dati integrati, ovvero un unico schema che sia indipendente dalle fonti e che rappresenti la base di dati che si vuole ottenere dall'integrazione.

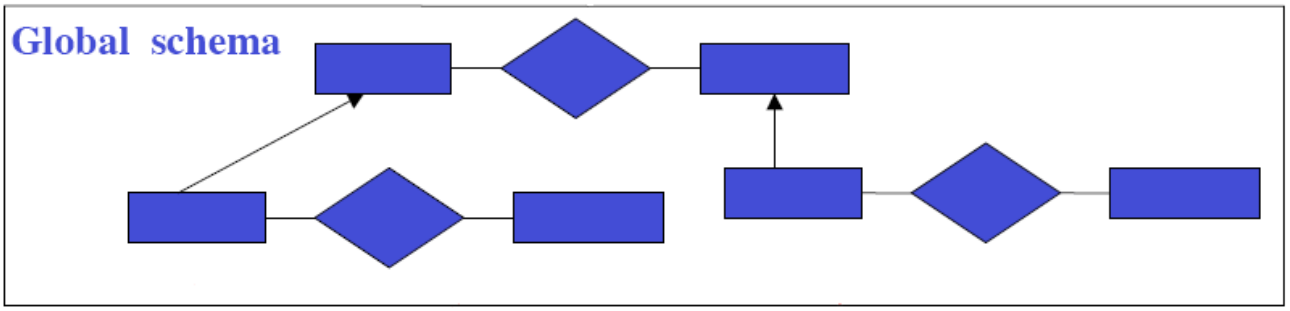

Figura 3 - Diagramma ER- Global Schema

La definizione del Global Schema viene fatta utilizzando il diagramma ER con le normali tecniche di modellazione concettuale delle basi di dati, in maniera tale da non perdere la capacità di espressione che si ha normalmente. Non necessariamente il Global Schema deve utilizzare tutti i dati presenti nelle fonti, ma solamente quelli di cui si ha bisogno per raggiungere lo scopo che ci si prefigge, allo stesso tempo, in fase di modellazione non ci si deve preoccupare di come fare a

recuperare le informazioni dalle sorgenti, in modo da non limitarsi nella definizione del Global Schema, con un altro passaggio poi si vedrà come portare i dati dalle fonti al quest'ultimo.

Il mapping

Allo scopo di generare un sistema automatico che consenta di portare i dati dalle fonti alla visione unica del global schema è necessario definire alcune regole sulle informazioni in maniera tale che queste possano essere ricostruite. Per questo proposito viene utilizzata una tecnica detta *mapping* che consiste nel creare uno strato di regole scritte in logica di prim'ordine che definiscano le relazioni tra le fonti ed il global schema. Definire un mapping adeguato è un passo fondamentale nella progettazione di un sistema di integrazione dei dati, in quanto dal buon uso delle tecniche di mapping non solo dipende l'efficienza della soluzione, ma anche l'applicabilità della stessa. Parleremo più approfonditamente del mapping in seguito.

Formalizzazione di un Data Integration System

Si definisce Data Integration System una tripla $\mathcal{I}=\{\mathcal{G}, \mathcal{S}, \mathcal{M}\}$ dove

- $\mathcal{G}$ è il global schema espresso in un linguaggio $\mathcal{L}_{\mathcal{G}}$ su un alfabeto $\mathcal{A}_{\mathcal{G}}$. L'alfabeto comprende un simbolo per ogni elemento di $\mathcal{G}$.
- $\mathcal{S}$ è il source schema espresso in un linguaggio $\mathcal{A}_{\mathcal{S}}$. L'alfabeto $\mathcal{A}_{\mathcal{S}}$ comprende un simbolo per ogni elemento di $\mathcal{S}$
- $\mathcal{M}$ è il mapping tra $\mathcal{G}$ ed $\mathcal{S}$, costituito da un insieme di asserzioni nella forma

$$q_{\mathcal{S}} \rightsquigarrow q_{\mathcal{G}}$$
$$q_{\mathcal{G}} \rightsquigarrow q_{\mathcal{S}}$$

Dove q_S e q_G sono query rispettivamente sul source schema e sul global schema. Le query q_S sono espresse in un linguaggio $\mathcal{L}_{\mathcal{M},S}$ sull'alfabeto $\mathcal{A}_S$, mentre le query q_G sono espresse in un linguaggio $\mathcal{L}_{\mathcal{M},G}$ sull'alfabeto $\mathcal{A}_G$.

In sostanza il source schema S descrive la struttura delle sorgenti dove sono i dati reali, mentre il global schema G descrive una vista riconciliata e integrata delle sorgenti.
Le query poste ad I sono poste in termini del global schema ed espresse in un linguaggio $\mathcal{L}_Q$ sull'alfabeto $\mathcal{A}_G$.

Semantica di un Data integration System

Database legale

Consideriamo un database sorgente per I come un database D che sia conforme al source schema S e che ne rispetti tutti i vincoli. Definiamo Global Database per I ogni database che sia conforme a G. Un generico database B è detto legale rispetto a D se:

- B è legale rispetto a G cioè soddisfa tutti i vincoli di G
- B soddisfa il mapping $\mathcal{M}$ rispetto a D.

La seconda asserzione si interpreta a seconda di quali siano le asserzioni del mapping $\mathcal{M}$. Questa definizione ci fa capire che esistono diversi database legali per il Data Integration System, e quindi gli studi sulla Data Integration devono correlarsi con quelli legati a database con informazione incompleta.

Semantica di una query su G

Una query di arità n è una formula con n variabili libere espressa in termini del global schema G. Detto DB un database, si denota con q^{DB} l'insieme delle tuple appartenenti a DB che soddisfano q. Dato un database sorgente D per I, la risposta $q^{I,D}$ ad una query q su I rispetto a D, è l'insieme di

tuple t di oggetti nell'alfabeto tali che $t \in q^B$ per ogni global database B che è legale per $\mathcal{I}$ rispetto a D; tale risposta è detta risposta certa. Il problema di trovare risposte certe è un problema di implicazioni logiche.

Il problema duale, è invece quello di trovare le *risposte possibili a q* cioè quelle tuple t tali che $t \in q^B$ per qualche global database B che è legale per $\mathcal{I}$ rispetto a D.

Il mapping

È evidente che un ruolo di grande importanza è ricoperto dalla scelta delle regole che definiscono il mapping tra il Global Schema e le sorgenti, ed è esattamente il modo in cui il mapping viene fatto che influenza il modo in cui le query saranno eseguite. Ci sono due approcci generali per descrivere il mapping, chiamati Local As View (LAV) e Global As View (GAV) che generano differenti vantaggi e svantaggi nel loro utilizzo. Nel seguito questi due approcci saranno trattati separatamente e sarà specificato per ognuna quali sono i vantaggi e gli svantaggi dell' utilizzo.

Local As View

Descrizione generale

Il concetto di base del metodo Local As View per il mapping è la possibilità di esprimere in termini del Global Schema tutte le sorgenti, ovvero:

detto $\mathcal{I}=\{\mathcal{G}, \mathcal{S}, \mathcal{M}\}$ un sistema di Data Integration, un mapping basato sull'approccio LAV associa ad ogni elemento s del source schema $\mathcal{S}$ una query $q_{\mathcal{G}}$ sul Global Schema $\mathcal{G}$.
In altre parole un mapping di tipo LAV è un insieme di asserzioni del tipo

$$s \rightsquigarrow q_{\mathcal{G}}$$

una per ogni elemento s di $\mathcal{S}$

Uno dei principali vantaggi dell'utilizzo di tale approccio è la possibilità, laddove ci sia un Global Schema ben definito e stabile, di estendere l'insieme delle sorgenti semplicemente aggiungendo alcune query nel mapping; di contro cambiare la struttura del Global Schema corrisponde a rimappare tutte le sorgenti. Uno degli svantaggi, è invece il fatto che la risposta ad una query risulta più complessa.

Accuratezza della conoscenza

Diversi studi sono stati condotti sulle tipologie di asserzioni che caratterizzano le sorgenti nel mapping LAV. In particolar modo si è cercato di classificare in che modo le sorgenti venivano specificate nel LAV. Ogni specifica di un elemento di s è denotata da as(s) e determina quanto è accurata la conoscenza della sorgente, in altre parole quanto i dati della sorgente soddisfano i criteri della vista $q_{\mathcal{G}}$ e quindi come debbano essere interpretati Si distinguono diversi tipi di viste:

- **Sound Views:** Quando una sorgente è sound, ogni sua tupla soddisfa i criteri di $q_{\mathcal{G}}$, ma il fatto che una tupla non faccia parte della sorgente non significa che essa non faccia parte di $q_{\mathcal{G}}$.

  Se as(s)=sound, dato un database sorgente $\mathcal{D}$, dal fatto che una tupla sia in $s^{\mathcal{D}}$ si può concludere che soddisfa la vista associata sul global schema, mentre se una tupla non da parte di $s^{\mathcal{D}}$ non si può dire che non soddisfi la vista associata. Formalmente, se as(s)=sound, un database B soddisfa s ~> $q_{\mathcal{G}}$ rispetto a $\mathcal{D}$ se

  $$s^{\mathcal{D}} \subseteq q^{\mathcal{B}}_{\mathcal{G}}$$

- **Complete views:** Quando una sorgente è complete il fatto che una tupla sia in $s^{\mathcal{D}}$ non significa che questa soddisfi i criteri di $q_{\mathcal{G}}$ mentre, se la tupla non fa parte di $s^{\mathcal{D}}$ sicuramente non soddisfa la query rispetto al global schema.

  Formalmente, se as(s)=complete, un database B soddisfa s ~> $q_{\mathcal{G}}$ rispetto a $\mathcal{D}$ se

  $$s^{\mathcal{D}} \supseteq q^{\mathcal{B}}_{\mathcal{G}}$$

- **Exact Views:** una sorgente s è esatta quando è Sound e Complete, ovvero l'insieme delle sue tuple è esattamente quello delle tuple che soddisfano $q_{\mathcal{G}}$. Formalmente, se as(s)=exact, un database B soddisfa s ~> $q_{\mathcal{G}}$ rispetto a $\mathcal{D}$ se

  $$s^{\mathcal{D}} = q^{\mathcal{B}}_{\mathcal{G}}$$

Global as view

Descrizione Generale

Nell'approccio GAV il mapping associa ad ogni elemento g del Global Schema $\mathcal{G}$ una query q_S sulle sorgenti S. Ogni entità del Global Schema viene quindi espressa come una asserzione in logica di primo ordine sulle sorgenti:

$$g \rightsquigarrow q_S$$

L'approccio GAV in un certo senso dice esplicitamente al Data Integration System dove recuperare i dati per la visione globale. Questo comporta uno svantaggio in termini di flessibilità della mappatura, perché all'aumento di una sorgente, corrisponde la rimappatura di una gran parte del Global Schema.

Accuratezza della conoscenza

La caratterizzazione del livello di conoscenza in una mappatura di tipo GAV, si misura non più sulle sorgenti, ma sul global schema. Come nel LAV si denotano 3 tipi di specificazioni: sound, complete e exact.

- **Sound Views:** se una specificazione as(g)=sound un database B soddisfa $g \rightsquigarrow q_S$ rispetto a $\mathcal{D}$ se

  $$q^{\mathcal{B}}_S \subseteq g^{\mathcal{D}}$$

  quindi se ogni tupla della query sulla sorgente è contenuta nell'elemento mappato del global schema. Il fatto che una tupla non sia nella query sulla sorgente però non significa che non possa far parte del global schema.

- **Complete views:** se una specificazione as(g)=complete un database B soddisfa $g \rightsquigarrow q_S$ rispetto a $\mathcal{D}$ se

$$q^{\mathcal{B}}_{S} \supseteq g^{\mathcal{D}}$$

quindi se ogni tupla appartenente ad un elemento g del global schema appartiene anche alla query sulla sorgente, ma non è necessariamente vero il viceversa.

- **Exact views:** una vista è detta exact (as(g)=exact) quando è sound e complete e quindi

$$q^{\mathcal{B}}_{S} = g^{\mathcal{D}}$$

Quindi quando ogni elemento appartenente alla query sulla sorgente appartiene all'elemento mappato del Global Schema g, e viceversa.

GLAV: Tra LAV e GAV

GLAV è un misto tra i due approcci sopra presentati, ovvero un mapping dove le relazioni sono espresse come asserzioni tra viste sul Global Schema e viste sulle sorgenti, nella seguente forma:

$$q_S \rightsquigarrow q_G$$

Dove q_S è una query sulle sorgenti e q_G è una query sul Global Schema, della stessa arità di q_S.
Dato un Database sorgente $\mathcal{D}$ un database B legale rispetto a $\mathcal{G}$ soddisfa $\mathcal{M}$ rispetto a $\mathcal{D}$ se

$$q^{\mathcal{B}}_{S} \subseteq q^{\mathcal{B}}_{G}$$

Altre Sfide della Data Integration

Molti altri sono gli ambiti di studio della Data Integration, alcuni vanno nella direzione del miglioramento di quanto presentato in questa tesina, altri studi si occupano di migliorare le risposte alle query presentate su un certo tipo di mapping. Altri aspetti altrettanto interessanti sono quelli dello Schema Matching e del Data Matching, ovvero del riconoscimento automatico di Schema o di tipologie di dati all'interno di sorgenti eterogenee.-

Integrazione dei dati nei sistemi informativi di un ente di formazione

In questa sezione della tesina, verrà mostrato il modello teorico per l'integrazione dei dati nei sistemi informativi di un ente di formazione. I passi che verranno fatti saranno i seguenti:

- Modellazione del Global Schema
- Descrizione delle attuali sorgenti
- Mapping LAV

I database rappresentati fanno riferimento ad un sistema reale utilizzato in un ente di formazione.

Global Schema

Al fine di integrare il sistema, è necessario proporre un Global Schema che sia indipendente dalle sorgenti, e sia legato solamente al problema evidenziato in Obiettivo e Motivazioni.
Alla fine del progetto, lo schema globale che si vuole ottenere è rappresentabile mediante il diagramma ER esposto in fig. 4.

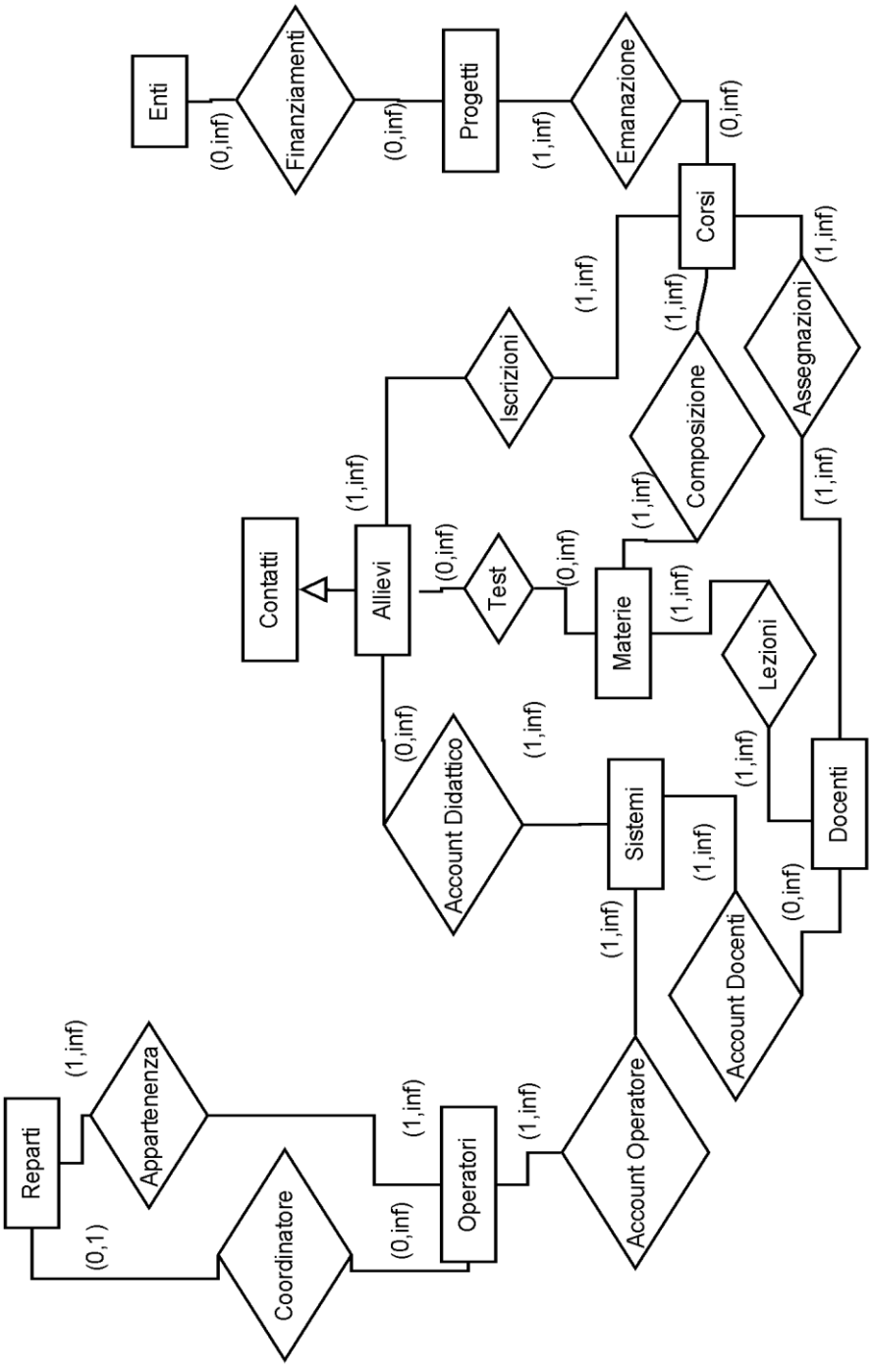

Figura 4 - Diagramma ER del Global Schema

L'entità **contatti**, corrisponde a tutte le persone che vengono contattate dai vari reparti, per diversi motivi quali, frequentazione di corsi, forniture di vario tipo, sondaggi o ricerche di mercato. Gli **allievi** sono una parte dei contatti, ed in particolare sono quei contatti che abbiano partecipato ad uno o più **corsi**. Gli allievi, a seconda del tipo di corso possono avere diversi **account didattici** per i **sistemi** informatici dell'azienda, siano essi interni od esterni.
Ogni contatto proviene da una **fonte** fondamentale da individuare per scopi di marketing.
Ogni corso è una **emanazione** di un **progetto** che fa capo ad un'asse di **finanziamento** tra quelli presentati dagli **enti** pubblici che bandiscono i soldi comunitari o nazionali.
Ogni **corso** è **assegnato** a dei **docenti** che hanno il loro **account docente** sui sistemi informatici dell'azienda.
Ogni corso è composto da **Materie** ed ogni **lezione** della materia è tenuta da un docente. Su ogni materia gli allievi affrontano dei **test**.
L'azienda è composta da **reparti** a cui appartengono **operatori** che hanno diversi **account operatore** sui sistemi aziendali.
Ogni reparto ha un **coordinatore**.

Modello Relazionale del Global Schema

In questa sezione si vanno a definire i vari attributi delle relazioni che caratterizzano il global schema. Si fa notare, che in alcuni casi la situazione è semplificata rispetto alle reali necessità aziendali, che spesso vengono sviluppate con relazioni da decine e decine di attributi l'una.

Contatti (AID**,** Nome, Cognome, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia, titoloDiStudio**)**

Allievi (AID**,** Nome, Cognome, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia, titoloDiStudio**)**

Sistemi(SID, Nome)

Corsi(CodiceCorso, Nome, Sede, PID)

Iscrizioni(CodiceCorso,AID)

Reparti (RID, Nome reparto, Coordinatore)

Operatori(OID, RID, Cognome, Nome, Costo Orario, Ruolo, DataDiNascita)

Docenti(DID, Nome, Cognome, eMail, telefono, Costo Orario, indirizzo, città, provincia, titoloDiStudio)

Enti(EID, Nome Ente)

Progetti(PID, Nome, Asse, Codice, valoreEuro, EID)

AccountDidattico(SID, AID, username, password, note)

AccountOperatore(SID, OID, username, password, note)

AccountDocenti(SID, DID, username, password, note)

Appartenenza(OID,RID**)**

Assegnazioni(CodiceCorso, DID**)**

Materie (MID, Nome**)**

Composizione(MID, CodiceCorso, Ore)

Lezioni(DID,MID, CodiceCorso, Durata, Data)

Test(MID, AID,CodiceCorso, Data, Voto**)**

Emanazione(PID,Codice Corso, ValoreEuro**)**

Coordinatore(OID, RID**)**

Definizione delle sorgenti

Le sorgenti sono presenti su diversi server dell'azienda, e sono di tipo eterogeneo, nel senso che alcune sono delle tabelle di database, altre dei file XML ed altre ancora dei file Excel esportati dai sistemi in questione, o creati appositamente per poter integrare dei dati altrimenti non disponibili.

L'accesso non è possibile a tutte le sorgenti, poiché alcune di queste sono alla base di software proprietari, i cui database sono protetti da password; e proprio per queste verranno utilizzati dei file esportati, oppure dei file che servono per simularli.

Le sorgenti vengono definite nelle seguenti tabelle, nelle quali viene anche specificato a volte come caricare i dati.

SugarCRM	
Tutti i contatti che l'ente ha vengono inseriti nel database di SugarCRM, ovvero un CRM Open Source che viene utilizzato per lo più dal reparto marketing di tale ente. In questo db, esiste un campo per evidenziare quali di questi contatti siano allievi	
Specifica della fonte	
Tipo:	Viste su Database MySQL
Relazioni:	**sugarCRM_contacts(**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**)**
Tabelle da estrarre:	Contacts, contacts_cstm
Calcolo SQL delle viste:	SELECT c.id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street,

	primary_address_city, primary_address_state, cstm.titolodistudio_c, cstm.allievo_c FROM contacts c, contacts_cstm cstm WHERE c.id=cstm.id_c ;

Xoops per allievi

Il portale aziendale, utilizza Xoops, che è un CMS Open Source, ed ogni allievo deve avere un account su quel portale, in modo da poter usufruire di servizi che sono riservati appunto agli iscritti ai corsi

Specifica della Fonte

Tipo:	Viste su Database MySQL
Relazioni:	**Xoops_users_pupils(**uid, name, uname, password, email**)**
Tabelle da estrarre:	Xoops_users
Calcolo SQL delle viste:	SELECT uid, name, uname, password, email from xoops_users u, xoops_groups g where (g.name=’allievi’)

Xoops per operatori

Il portale aziendale contiene una parte interna dedicata solamente agli operatori. Questi fanno quindi parte degli utenti del portale, ed in particolare sono quelli assegnati al gruppo “operatori”. Possono inoltre essere estratti i vari reparti (come parte dei gruppi) e sono tutti quei gruppi che non si chiamano “operatori”,”anonimi”,“registrati” o “webmaster”.

Specifica della Fonte

Tipo:	Viste su Database MySQL
Relazioni:	**Xoops_users_employees(**uid, name, uname, password, email, groupid**)**

	Xoops_departments(groupid, name)
Tabelle da estrarre:	Xoops_users xoops_groups xoops_users_groups_link
Calcolo SQL delle viste:	**Xoops_users_view**: SELECT uid, name, uname, password, email from xoops_users u, xoops_groups g, xoops_users_groups_link l where g.name=”operatori” and l.groupid=g.groupid and u.uid=l.uid **xoops_departments_view:** SELECT groupid, name from xoops_groups where NOT (name =’operatori’ or name=’anonimi’ or name=’registrati’ or name=’webmaster’)

Moodle

È il Learning Management System utilizzato dall’ente, e sul quale ogni allievo deve avere una iscrizione per accedere ai contenuti ed al materiale didattico relativi ai corsi.

Ogni allievo è iscritto ad uno o più corsi, e effettua i suoi test collegando i risultati ad un corso, e non ad una materia. Imponendo però una naming convention (diciamo con una funzione di hashing h) sui nomi dei quiz, correlandoli alle materie insegnate, si può ottenere una relazione tra i test e le materie.

Specifica della Fonte	
Tipo:	Viste su Database MySQL
Relazioni:	**moodle_users(**id, username, password, email, firstname, lastname**)** **moodle_tests(**qid, courseId, subject, userId,sumgrade, timestart**)**
Tabelle da estrarre:	Mdl_users

Calcolo SQL delle viste:	**Moodle_users:** SELECT id, username, password, email, firstname, lastname, group from mdl_users **Moodle_test:** SELECT a.id, a.userId, h(a.name) as subject, a.sumgrade, a timestart q.course as courseId from mdl_quiz_attempts a, mdl_quiz WHERE a.qid=q.id

Filezilla server

FileZilla Server il server FTP Open Source per i corsi che trattano di sviluppo web. Contiene gli account che gli allievi di quei corsi utilizzano per poter accedere ai propri spazi e pubblicare i propri lavori.

Specifica della Fonte

Tipo:	File XML
Relazioni:	**filezilla_server_users(**username, password, group**)**
Note:	o Lo username è dato dalle prime 5 lettere del cognome di un allievo + le prime 2 del nome. Ad es. Roberto Saviano avrà username *saviaro* . o Inoltre, ogni gruppo corrisponde ad un corso, ed ha lo stesso nome del codice del corso presente in Proforma che è il sistema di base per la gestione dei corsi di formazione.

Proforma

È un programma che contiene tutta la parte gestionale dei corsi di formazione, quindi informazioni su tutti i corsi, gli allievi, gli enti e i docenti.

È un software proprietario, e la password del database, non è di conoscenza dell'ente, quindi si devono necessariamente utilizzare due file Excel estratti dal sistema

Specifica della Fonte	
Tipo:	File Excel
Relazioni:	**proforma_allievi(**id, firstName, lastName, courseId**)** **proforma_corsi(**courseId, name, place,PID**)** **proforma_argomenti(**subjId, name**)** **proforma_assegnazioni(**courseId,docId**)** **proforma_lezioni(**subjId, courseId, docId, lenght, data**)** **proforma_composizione**(courseId, subjId, hours**)** **proforma_docenti(**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree**)** **proforma_progetti**(projCode, name, axe, value, agencyId**)** **proforma_enti(**agencyId, name**)** **proforma_operatori(**operatoreID, name, surname, DOB, placeofbirth, role, cost**)**
Note	○ Da queste tabelle si può avere la chiave per ricostruire I legami tra le varie relazioni.

Terminal Server Didattico

Sono gli account degli allievi sui diversi server di Desktop Remoto di Microsoft Windows 2003, al database del quale non è possibile accedere, si userà pertanto un file Excel che ne simuli la struttura.

Specifica della Fonte	
Tipo:	File Excel
Relazioni:	**ts_users_school(**username, password, group, firstName, lastName**)**
Note	○ Lo username è dato dalle prime 5 lettere del cognome di un utente e le prime 2 del nome. Ad es. Roberto Saviano avrà username *saviaro* .

	o Inoltre, ogni gruppo corrisponde ad un corso, ed ha lo stesso nome del codice del corso presente in Proforma.

Terminal Server Operativo

Sono gli account degli operatori sui diversi server di Desktop Remoto di Microsoft Windows 2003, al database del quale non è possibile accedere, si userà pertanto un file Excel che ne simuli la struttura.

Specifica della Fonte

Tipo:	File Excel
Relazioni:	**ts_users_ops(**username, password, group, firstName, lastName**)**
Note	o Lo username è dato dalle prime 5 lettere del cognome di un utente e le prime 2 del nome. Ad es. Roberto Saviano avrà username *saviaro* . o Inoltre ad ogni reparto corrisponde un gruppo di utenti, con determinati permessi di accesso alle cartelle. La relazione univoca tra utente e reparto la si trova su un portale, gestito dalla fonte Xoops

Elenco Sistemi

È necessario creare una fonte che elenchi i sistemi, e ad essi associ un chiave primaria, pertanto si potrebbe pensare di usare un file Excel contenente le informazioni minime utili per il nostro scopo, ovvero nome e chiave

Specifica della Fonte

Tipo:	File Excel
Relazioni:	**systems(**SID,name**)**
Note	

Mapping LAV

Il mapping utilizzato sarà di tipo LAV, poiché data la rapidità di cambiamento delle fonti nell'ente trattato, deve esserci la semplicità di aggiornamento dello stesso.

Verranno quindi specificate le sorgenti rispetto al Global Schema, per facilitare la lettura sono state utilizzate delle lettere maiuscole per le relazioni appartenenti al GS e minuscole per quelle appartenenti alle sorgenti.

- **sugarCRM_contacts(**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**)** → **{**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c **| Contatti(**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c**)}**

- **xoops_users_pupils(**uid,name,uname,password,email**)** : → **{**name,uname,password,email **| AccountDidattici(**'Xoops', AID, uname, password, note**)**∧ **Allievi (**AID**,** token*(name,1), token(name,2), dataDiNascita, telefonoCasa, telefonoMobile, email, indirizzo, città, provincia, titoloDiStudio**)}**

- **xoops_users_employees(**uid,name,uname,password,email,groupid**)** : → **{**name,uname,password,email **| AccountOperatore(**'Xoops', OID, uname, password, note**)**∧ **Operatori (**OID**,** RID, token(name,1), token(name,2), costoOrario, Ruolo, dataDiNascita, CostoOrario**)}**

- **xoops_departments(**groupid, name)→{groupid,name | **Reparti**(groupid, name, coordinatore)}

*Per Token(s,n) si intende una funzione che ritorna la n-esima parola da una stringa s

- **moodle_users(**id,username,password,email,firstname,lastname**) : → {**username,password,email,firstname,lastname | (**AccountDidattico(**'moodle', AID, username, password, note**))**∧ **Allievi (**AID**,** firstname, lastname, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia,titoloDiStudio**))** □ ((**AccountDocenti(**'moodle', DID, username, password, note**)**∧ **Docenti (**DID**,** firstname, lastname, eMail, indirizzo, città, provincia,titoloDiStudio**)) }**

- **moodle_tests(**qid, courseId, subject, userId, sumgrade, timestart**) :→ {**qid, courseId, subject, userId, sumgrade, timestart | **moodle_users(**userId,username,password,email,firstname,lastname**)** ∧ **AccountDidattico(**'moodle', AID, username, password, note**)** ∧ **Allievi (**AID**,** firstname, lastname, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia,titoloDiStudio**)** ∧

Corsi(courseId, Nome, Sede,PID**)** ∧ **Iscrizioni (**courseId, AID**)** ∧ **Materie (**subject, NomeMateria**)** ∧ **Composizione(**subject, CourseId, Ore**)** ∧ **Test(**subject, userId,CourseId, timestart, sumgrade**)}**

- **filezilla_server_users(**username, password, group**):→ {**username, password, group | **AccountDidattici(**'filezilla', AID, username, password, note**)** ∧ **Allievi (**AID**, N**ome, Cognome, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia,titoloDiStudio**)** ∧ **Iscrizioni (**group, AID**)** ∧ **Corsi (**group, Nome, Sede,PID**)}**

- **proforma_allievi(**id,firstName,lastName,courseId**) :→ {** firstName,lastName,courseId | **Allievi (**AID**,** firstName, lastName, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia, titoloDiStudio**)** ∧ **Corsi (**courseId, Nome, Sede,PID**)** ∧ **Iscrizioni (**courseId, AID**)**

- **proforma_corsi(**courseId, name, place,PID**) : → {** courseId, name, place,PID **|** **Corsi (**courseId, name, place,PID**)** ∧ **Progetti(PID**, Nome, Asse, Valore, IdEnte) **}**

- **proforma_docenti(**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree) **: → {** id, firstName, lastName, cost, address, city, region, studyDegree **|** **Docenti (**id, lastName, firstName, eMail, phone, cost, address, city, region, studyDegree**) }**

- **proforma_progetti**(projCode, name, axe, value, agencyId**): → {** projCode, name, axe, value, agencyId | **Progetti(**projCode, name, axe, value, agencyId) ∧ **Enti(**agencyId, AgencyName) }

- **proforma_enti(**agencyId, name**):→ {**agencyId, name | **Enti(**agencyId, name)}

- **proforma_operatori(**operatoreID, name, surname, DOB, placeofbirth, role, cost**):→ {** operatoreID, name, surname, DOB, placeofbirth, role, cost | **Operatori (**operatoreID**,** RID, surname, name, cost, role, DOB, placeofbith**)}**

- **proforma_argomenti(**subjId**,** name**) :→{**subjId, name | **Materie (**subjId, NomeMateria**) }**

- **proforma_assegnazioni(**courseId,docId**):→{** courseId, docId | **Corsi (**courseId, Nome, Sede,PID) ∧ **Docenti (**docId, lastName, firstName, eMail, phone, cost, address, city, region, studyDegree**)}**

- **proforma_lezioni(**subjId, courseId, docId, length, date**) :→{**subjId, courseId, docId, length, data | **Corsi (**courseId, Nome, Sede,PID) ∧ **Docenti(**docId, lastName,firstName, eMail, phone, cost, address, city, region, studyDegree**)** ∧ **Materie(**subjId, NomeMateria**)** ∧ **Lezioni(**docId, subjId, courseId, length, date**)**

- **proforma_composizione**(courseId, subjId, hours**):→{**courseId, subjId | **Corsi (**courseId, Nome, Sede,PID) ∧ **Materie(**subjId, NomeMateria**)** ∧ **Composizione(**subjId, courseId, hours**)}**

- **ts_users_school(**username, password, group, firstName, lastName**) : → {** username, password, firstName, lastName, group **|** **Account_Didattico(**'ts_school', AID, username, password, note**))** ∧ **Corsi (**group, Nome, Sede**)** ∧ **Iscrizioni (**group, AID**)** ∧ **Allievi (**AID**,** firstName, lastName, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincia, titoloDiStudio**)**

- **ts_users_ops(**username, password, group, firstName, lastName**) : → {** username, password, firstName, lastName, group **|** **Account_Operatori(**'ts_ops', OID, username, password, note**))** ∧ **Reparti(**RID, group, Coordinatore) ∧ **Operatori (**OID**,** RID, lastName, firstName, cost, role, DOB, placeofbirth**)}**

- **systems(**SID, name**):→ {**SID, Name **| Sistemi (**SID, name**)}**.

Trasformazione del mapping LAV in GAV

Nell'ambito di questa tesina è stato scelto un mapping di tipo LAV per via della variabilità delle sorgenti coinvolti e per la duttilità che LAV offre nell'aggiungerne di nuove o modificarle. Tuttavia il LAV, è un tipo di mapping per il quale la risposta alle query risulta complessa, mentre il GAV pur essendo più rigido sotto l'aspetto della modificabilità delle sorgenti, rende più semplice la risposta alle query. È possibile passare in qualche modo da LAV a GAV senza perdere la capacità di ottenere gli stessi risultati dalle query (query preservino transformation)? Sì, se il Global Schema ammette vincoli di integrità.

Come funziona

Per trasformare un mapping da LAV in GAV è necessario eseguire i seguenti passi:

- Innanzitutto si riscrive il mapping in modo tale che ogni variabile appaia in ogni atomo al massimo una volta.
- Il Global Schema G' è ottenuto da G introducendo
  - Una nuova relazione image_s/n per ogni relazione s/n in S
  - Una nuova relazione expand_s/(n+m) per ogni relazione s/n in S dove m è il numero delle variabili non distinte di $\rho_G(s)$. Assumiamo che le variabili in $\rho_G(s)$ siano enumerate come $Z_1,\ldots,Zn+m$ dove le variabili fino a Zn sono quelle distinte
  - Aggiungendo le seguenti dipendenze
    - Image_s[1,….,n] $\subseteq$ expand_s[1,…,n]
  - Per ogni relazione s in S e per ogni atomo $g(Zi_1,\ldots,Zik)$ occorrente in $\rho_G(s)$, aggiungere la dipendenza di inclusione
    - Expand_s[$i_1,\ldots, i_k$] $\subseteq$ g[1,….,k]
  - Per ogni relazione s in S e per ogni atomo $Z_i=Z_j$ occorrente in $\rho_G(s)$ aggiungere la simple equality gene rating dependency expand_s→ i=j
- Il nuovo mapping GAV è costituito da:
  - $\rho_S(image\_s)$:- s per ogni s$\in$ S

Trasformazione del mapping sviluppato

Al Global Schema per effettuare la trasformazione vanno inserite:

- Le immagini delle sorgenti
- Le espansioni delle sorgenti
- I vincoli d'integrità aggiunti dalla trasformazione

A seguire vedremo tutte le componenti da aggiungere al global schema

Immagini delle sorgenti

- **Image_sugarCRM_contacts(**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**)**
- **Image_Xoops_users_pupils(**uid, name, uname, password, email**)**
- **Image_Xoops_users_employees(**uid, name, uname, password, email, groupid**)**
- **Image_Xoops_departments**(groupid, name)
- **Image_moodle_users(**id, username, password, email, firstname, lastname**)**
- **Image_moodle_tests(**qid, courseId, subject, userId,sumgrade, timestart**)**
- **Image_filezilla_server_users(**username, password, group**)**
- **Image_proforma_allievi(**id, firstName, lastName, courseId**)**
- **Image_proforma_corsi(**courseId, name, place,PID**)**
- **Image_proforma_argomenti(**subjId, name**)**

- **Image_proforma_assegnazioni(**courseId,docId**)**
- **Image_proforma_lezioni(**subjId, courseId, docId, lenght, data**)**
- **Image_proforma_composizione**(courseId, subjId, hours**)**
- **Image_proforma_docenti(**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree**)**
- **Image_proforma_progetti**(projCode, name, axe, value, agencyId**)**
- **Image_proforma_enti(**agencyId, name**)**
- **Image_proforma_operatori(**operatoreID, name, surname, DOB, placeofbirth, role, cost**)**
- **Image_ts_users_school(**username, password, group, firstName, lastName**)**
- **Image_ts_users_ops(**username, password, group, firstName, lastName**)**
- **Image_systems(**SID,name**)**

Espansioni delle sorgenti

- **Expand_ sugarCRM_contacts/12(**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**)***
- **Expand_Xoops_users_pupils/13(**uid, name, uname, password, email,note,dataDiNascita,telefonoCasa,telefonoMobile,,indirizzo,città,provincial,titolodistudio**)**
- **Expand_Xoops_users_employees/10(**uid, name, uname, password, email, groupid,note, RID, costoOrario, Ruolo, dataDiNascita**)**
- **Expand_Xoops_departments/3**(groupid, name,coordinatore)

- **Expand_moodle_users/16(**id, username, password, email, firstname, lastname, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città provincia, titoloDiStudio,AID,DID**)**
- **Expand_moodle_tests/27(**qid, courseId, subject, userId,sumgrade, timestart, username, password, email, firstname, lastname, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città provincia, titoloDiStudio, Nome,Sede, PID, Nome Materia, Ore, AID, timestart, sumgrade**)**
- **Expand_filezilla_server_users/18(**username, password, Nome, Cognome, group, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città, provincia, titoloDiStudio,NomeCorso, Sede,PID,AID,eMail**)**
- **Expand_proforma_allievi/16(**id, firstName, lastName, courseId, AID, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincial, titoloDiStudio, nomeCorso, sede, PID**)**
- **Expand_proforma_corsi/8(**courseId, name, place,PID, nomeProgetto, asse, Valore, idEnte**)**
- **Expand_proforma_docenti/10(**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree**)**
- **Expand_proforma_progetti/5**(projCode, name, axe, value, agencyId, AgencyName**)**
- **Expand_proforma_enti/2(**agencyId, name**)**
- **Expand_proforma_operatori/8(**operatoreID, RID, name, surname, DOB, placeofbirth, role, cost**)**
- **Expand_proforma_argomenti/2(**subjId, name**)**
- **Expand_proforma_assegnazioni/14(**courseId,docId, Nome, Sede, PID, lastName, firstName, eMail, phone, cost, address, city, region, studyDegree**)**

- **Expand_proforma_lezioni/18(**subjId, courseId, docId, lenght, data, Nome, Sede, PID, lastName, firstName, eMail, phone, cost, address, city, region, studyDegree, NomeMateria**)**
- **Expand_proforma_composizione/7**(courseId, subjId, hours, Nome, Sede, PID, NomeMateria**)**
- **Expand_ts_users_school/19(**username, password, group, firstName, lastName, AID, note, nomeCorso, Sede, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincial, titoloDiStudio, note ,'const'**)**
- **Expand_ts_users_ops/14(**username, password, group, firstName, lastName, OID, note, RID, Coordinatore, cost, role, DOB, placeOfBirth, 'const'**)**
- **Expand_systems/2(**SID,name**)**

Dipendenze di inclusione generate dalla trasformazione

- **Image_sugarCRM_contacts[**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**]**⊆ **Expand_ sugarCRM_contacts[**id, first_name, last_name, birthdate, phone_home, phone_mobile, eMail1, primary_address_street, primary_address_city, primary_address_state, titolo_di_studio_c, allievo_c**]**
- **Image_Xoops_users_pupils[**uid, name, uname, password, email**]** ⊆ **Expand_Xoops_users_pupils[**uid, name, uname, password, email**]**
- **Image_Xoops_users_employees[**uid, name, uname, password, email, groupid**]** ⊆ **Expand_Xoops_users_employees[**uid, name, uname, password, email, groupid**]**
- **Image_Xoops_departments[**groupid, name]⊆ **Expand_Xoops_departments**[goupid, name]

- **Image_moodle_users[**id, username, password, email, firstname, lastname**]** ⊆ **Expand_moodle_users[**id, username, password, email, firstname, lastname, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città provincia, titoloDiStudio**]**
- **Image_moodle_tests[**qid, courseId, subject, userId,sumgrade, timestart**]** ⊆ **Expand_moodle_tests[**qid, courseId, subject, userId,sumgrade, timestart, username, password, email, firstname, lastname, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città provincia, titoloDiStudio, Nome,Sede, PID, Nome Materia, Ore**]**
- **Image_filezilla_server_users[**username, password, group**]** ⊆**Expand_filezilla_server_users[**username, password, group, note, dataDiNascita, telefonoCasa, telefonoMobile, indirizzo, città provincia, titoloDiStudio,NomeCorso, Sede,PID**]**
- **Image_proforma_allievi[**id, firstName, lastName, courseId**]** ⊆ **Expand_proforma_allievi[**id, firstName, lastName, courseId, AID, dataDiNascita, telefonoCasa, telefonoMobile, eMail, indirizzo, città, provincial, titoloDiStudio, nomeCorso, sede, PID**]**
- **Image_proforma_corsi[**courseId, name, place,PID**]** ⊆ **Expand_proforma_corsi[**courseId, name, place,PID]
- **Image_proforma_argomenti[**subjId, name**]** ⊆ **Expand_proforma_argomenti[**subjId, name**]**
- **Image_proforma_assegnazioni[**courseId,docId**]** ⊆ **Expand_proforma_assegnazioni[**courseId,docId]
- **Image_proforma_lezioni[**subjId, courseId, docId, lenght, data**]** ⊆ **Expand_proforma_lezioni[**subjId, courseId, docId, lenght, data]
- **Image_proforma_composizione**[courseId, subjId, hours**]** ⊆ **Expand_proforma_composizione**[courseId, subjId, hours]
- **Image_proforma_docenti[**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree**]** ⊆ **Expand_proforma_docenti[**id,firstName,lastName, eMail, phone, cost, address, city, region, studyDegree**]**
- **Image_proforma_progetti**[projCode, name, axe, value, agencyId**]** ⊆ **Expand_proforma_progetti**[projCode, name, axe, value, agencyId]
- **Image_proforma_enti[**agencyId, name**]** ⊆ **Expand_proforma_enti[**agencyId, name**]**

- **Image_proforma_operatori[**operatoreID, name, surname, DOB, placeofbirth, role, cost**]** ⊆ **Expand_proforma_operatori[**operatoreID, name, surname, DOB, placeofbirth, role, cost**]**
- **Image_ts_users_school[**username, password, group, firstName, lastName**]** ⊆ **Expand_ts_users_school[**username, password, group, firstName, lastName]
- **Image_ts_users_ops[**username, password, group, firstName, lastName**]** ⊆ **Expand_ts_users_ops[**username, password, group, firstName, lastName]
- **Image_systems[**SID,name**]** ⊆ **Expand_systems[**SID,name**]**

Altre dipendenze d'inclusione

- **Expand_ sugarCRM_contacts[**1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] ⊆ **Contatti[**1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11**]**
- **Expand_Xoops_users_pupils[**1, 2, 5, 7, 8, 9, 10, 11, 12, 13**]** ⊆ **Allievi [**2, 3, 7, 4, 5, 6, 8, 9, 10, 11**]**
- **Expand_Xoops_users_pupils[**3,4**]** ⊆**AccountDidattici[**3, 4**]**
- **Expand_Xoops_users_employees[**2, 2, 10, 11**]** ⊆ **Operatori [**token1[2],token2[2], 5,7]
- **Expand_Xoops_users_employees[**3,4**]** ⊆ **AccountOperatore**[2,3]
- **Expand_Xoops_departments**[1,2,3] ⊆ **Reparti**[1,2,3]}
- **Expand_moodle_users[**2, 3, 15**]** ⊆ **AccountDidattico[**3,4,2**]**

- **Expand_moodle_users[**4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15] ⊆ **Allievi [7,** 2, 3, 4, 5, 6, 8, 9,10,11 , 1**]**
- **Expand_moodle_users[**4, 5, 6, 11, 12, 13, 14,16**]** ⊆ **Docenti [4,** 2, 3, 5, 6, 7, 8, 1**]**
- **Expand_moodle_users[**2, 3, 15**]** ⊆ **AccountDocenti [**3,4,2**]**
- **Expand_moodle_tests[7**, 8, 25**]** ⊆ **AccountDidattico[**3, 4, 2**]**
- **Expand_moodle_tests[**9, 10, 11, 13, 14, 15, 16, 17, 18, 19,25**]** ⊆ **Allievi [**7**,** 2, 3, 4, 5, 6, 8, 9, 10, 11,1**]**
- **Expand_moodle_tests[**2**,** 20, 21, 22**]** ⊆ **Corsi[**1**,** 2, 3, 4**]**
- **Expand_moodle_tests[**2**,** 25**]** ⊆ **Iscrizioni**[1, 2]
- **Expand_moodle_tests[**3**,** 23**]** ⊆ **Materie [**1, 2**]**
- **Expand_moodle_tests[**3, 2, 24**]** ⊆ **Composizione[**1, 2, 3**]**
- **Expand_moodle_tests[**3, 4, 2, 26,27**]** ⊆ **Test[**1,2,3,4,5**]**
- **Expand_filezilla_server_users[**1, 2, 6, 17**]** ⊆ **AccountDidattici[**3, 4, 5, 2**]**
- **Expand_filezilla_server_users[**3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 18**]** ⊆ **Allievi [**2, 3, 4, 5, 6, 7, 8, 9, 10,11,1**]**
- **Expand_filezilla_server_users[**5, 17**]** ⊆ **Iscrizioni [**1, 2**]**
- **Expand_filezilla_server_users[**5, 14,15,16] ⊆ **Corsi [**1,2,3,4**]**
- **Expand_proforma_allievi[**2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13**]** ⊆ **Allievi [**2, 3, 1, 4, 5, 6, 7, 8, 9, 10, 11**]**
- **Expand_proforma_allievi[**4, 14, 15, 16] ⊆ **Corsi [**1, 2, 3, 4**]**
- **Expand_proforma_allievi[**4, 5] ⊆ **Iscrizioni [**1, 2**]**

- **Expand_proforma_corsi[**1, 2, 3, 4**]** ⊆ **Corsi [**1, 2, 3, 4**]**
- **Expand_proforma_corsi[**4, 5, 6, 7, 8**]** ⊆ **Progetti[**1, 2, 3, 4, 5]
- **Expand_proforma_docenti[**1, 2, 3, 4, 5, 6, 7, 8, 9, 10**]** ⊆ **Docenti [**1, 2, 3, 4, 5, 6, 7, 8, 9, 10**]**
- **Expand_proforma_progetti**[1, 2, 3, 4, 5**]** ⊆ **Progetti[**1, 2, 3, 4, 5]
- **Expand_proforma_progetti**[5, 6**]** ⊆ **Enti[**1, 2]
- **Expand_proforma_enti[**1, 2**]** ⊆ **Enti[**1, 2]
- **Expand_proforma_operatori[**1,2,3,4,5,6,7**]** ⊆ **Operatori [**1,2,3,4,7,8,5,6**]**
- **Expand_proforma_argomenti[**1, 2**]** ⊆ **Materie[**1, 2**]**
- **Expand_proforma_assegnazioni[**1, 3, 4, 5**]** ⊆ **Corsi [**1, 2, 3, 4]
- **Expand_proforma_assegnazioni[**2, 6, 7, 8, 9, 10, 11, 12, 13, 14**]** ⊆ **Docenti [**1, 2, 3, 4, 5, 6, 7, 8, 9, 10**]**
- **Expand_proforma_lezioni[**2, 6, 7, 8**]** ⊆ **Corsi** [1, 2, 3, 4]
- **Expand_proforma_lezioni[**3, 9, 10, 11, 12, 13, 14, 15, 16, 17**]** ⊆ **Docenti[**1,2, 3, 4, 5, 6, 7, 8, 9, 10**]**
- **Expand_proforma_lezioni[**1, 18**]** ⊆ **Materie[**1, 2**]**
- **Expand_proforma_lezioni[**3, 1, 2, 4, 5] ⊆ **Lezioni[**1, 2, 3, 4, 5**]**
- **Expand_proforma_composizione/7**[courseId, subjId, hours, Nome, Sede, PID, NomeMateria**]**
- **Expand_proforma_composizione**[1, 4, 5, 6**]** ⊆ **Corsi [**1, 2, 3, 4]
- **Expand_proforma_composizione**[2, 7**]** ⊆ **Materie[**1, 2**]**

- **Expand_proforma_composizione**[2, 1, 3**]** ⊆ **Composizione[**1, 2, 3**]**
- **Expand_ts_users_school[**1, 2, 6, 7, 18**]** ⊆ **Account_Didattico[**3,4,2,5,1**]**
- **Expand_ts_users_school[**3, 6**]** ⊆ **Iscrizioni [**1, 2**]**
- **Expand_ts_users_school[**6, 4, 5,10,11,12,13,14,15,16,17**]** ⊆ **Allievi [**1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11**]**
- **Expand_ts_users_ops[**14, 6, 1, 2, 7**]** ⊆ **Account_Operatori[**1, 2, 3, 4, 5**]**
- **Expand_ts_users_ops[**8, 3, 9**]** ⊆ **Reparti[**1, 2, 3]
- **Expand_ts_users_ops[**6, 8, 5, 4, 10, 11, 12, 13**]** ⊆ **Operatori [**1, 2, 3, 4, 5, 6, 7, 8]
- **Expand_systems[**1, 2**]** ⊆ **Sistemi [**1, 2**]**

Mapping Risultante

Una volta trasformato il Global Schema, possiamo descrivere il nuovo Mapping GAV.

- **Image_sugarCRM_contacts :- sugarCRM_contacts**
- **Image_Xoops_users_pupils :- Xoops_users_pupils**
- **Image_Xoops_users_employees :- Xoops_users_employees**
- **Image_Xoops_departments**:- **Xoops_departments**
- **Image_moodle_users:- moodle_users**
- **Image_moodle_tests:- moodle_tests**
- **Image_filezilla_server_users:- filezilla_server_users**

- **Image_proforma_allievi:- proforma_allievi**
- **Image_proforma_corsi:- proforma_corsi**
- **Image_proforma_argomenti:- proforma_argomenti**
- **Image_proforma_assegnazioni:- proforma_assegnazioni**
- **Image_proforma_lezioni:- proforma_lezioni**
- **Image_proforma_composizione**:- **proforma_composizione**
- **Image_proforma_docenti:- proforma_docenti**
- **Image_proforma_progetti**:- **proforma_progetti**
- **Image_proforma_enti:- proforma_enti**
- **Image_proforma_operatori:- proforma_operatori**
- **Image_ts_users_school:- ts_users_school**
- **Image_ts_users_ops:- ts_users_ops**
- **Image_systems:- systems**

Conclusioni e annotazioni

Sebbene lo schema presentato sia semplificato per la dimensione orizzontale delle relazioni, rispetto al sistema completo dell'azienda, questo lavoro copre tutti gli ambiti nei quali sia necessario integrare le informazioni per poter avere un più fluido trattamento delle informazioni da parte dell'impresa oggetto di studio.

Il cambiamento delle leggi a cui tale azienda è sottoposta, la varietà dei progetti sui quali opera, l'altissimo turnover di risorse e la predilezione della dirigenza per la sperimentazione di nuovi prodotti informatici [possibilmente Open Source) generano un proliferare di nuovi sistemi e nuovi processi, che portano la disgregazione dell'informazione ad aumentare a vista d'occhio, e rendono obsoleto o incompleto il Global Schema qui presentato nell'arco di uno o due anni, sebbene questo rappresenti ad oggi completamente l'attività centrale dell'ente.

Da quando il progetto è partito, l'esigenza di integrazione si fa sempre più forte, vista la complessità del mercato nel quale l'azienda opera, tuttavia la realizzazione del progetto sembra tuttora essere troppo dispendiosa sia dal punto di vista temporale che economico, anche considerando il fatto che per i motivi sopraccitati l'operazione andrebbe ripetuta o integrata ogni due anni circa.

Ad ogni modo nonostante la fase di erogazione del progetto non sia mai stata avviata, questo documento viene preso come base per un futuro sviluppo nell'ottica dell'integrazione dei dati.

Bibliografia

- A.Calì, D.Calvanese, G.De Giacomo, M.Lenzerini – "**On the Expressive Power of Data Integration Systems**"
- M.Lenzerini – "**Data Integration: A Theoretical Perspective**"
- J.D.Ullman – "**Information integration using logical views**
- R.Rosati, G.De Giacomo – "**Logic-based information integration**"