

**Università degli Studi di Modena e Reggio Emilia**

**Facoltà di Ingegneria di Modena**

---

Corso di Laurea Triennale in Ingegneria Informatica – *Nuovo Ordinamento*

**FASHION IMPORTER: PROGRAMMA PER  
L'IMPORTAZIONE DATI PER L'INDUSTRIA DI  
ABBIGLIAMENTO**

Relatore:

Prof. Sonia Bergamaschi

Correlatore:

Ing. Roberto Pellegrino

Candidato:

Luca Torreggiani

---

Anno Accademico 2008/2009



*ai miei genitori*

## INDICE

<b>1. INTRODUZIONE</b> .....	- 5 -
<b>2. IL SISTEMA INFORMATIVO AZIENDALE</b> .....	- 7 -
2.1 ERP .....	- 10 -
2.2 Legacy System .....	- 11 -
2.2.1 Caratteristiche di un Legacy System .....	- 11 -
2.2.2 Tipologie di Legacy System .....	- 12 -
2.2.3 Trattamento dei Legacy System .....	- 13 -
<b>3. IL SISTEMA INFORMATIVO DI SPAZIOSEI</b> .....	- 15 -
3.1 L'ERP .....	- 16 -
3.2 Nuove esigenze di business .....	- 18 -
3.3 Da DB2 a MySQL .....	- 21 -
3.3.1 Panoramica generale DB2 .....	- 21 -
3.3.2 Panoramica e storia di MySQL .....	- 22 -
3.3.3 Caratteristiche di MySQL.....	- 22 -
3.3.4 Vantaggi di MySQL 5.0 .....	- 23 -
<b>4. PROCESSO DI IMPORTAZIONE</b> .....	- 24 -
4.1 Base di dati Legacy .....	- 25 -
4.1.1 Anagrafica prodotti .....	- 25 -
4.1.2 Descrizione base di dati ERP.....	- 26 -
4.2 Base di dati MySQL.....	- 31 -
4.2.1 Diagramma ER.....	- 32 -
4.3 Estrazione dati.....	- 35 -
4.4 Definizione della tabella di mapping .....	- 39 -
4.5 Implementazione della procedura Java di importazione dati	- 43 -
4.6 Strumenti utilizzati.....	- 44 -
4.6.1 DBVisualizer.....	- 44 -
4.6.2 Kettle.....	- 45 -
4.6.3 MySQL Administrator e MySQL Query Browser.....	- 46 -
4.6.4 Il framework Eclipse .....	- 47 -
4.6.5 Linguaggio di programmazione.....	- 48 -

**5. CONCLUSIONI E SVILUPPI FUTURI..... - 49 -**  
**6. BIBLIOGRAFIA ..... - 52 -**

# 1. INTRODUZIONE

Il progetto di tesi tratta l'implementazione di una procedura standard di importazione dati; la realtà aziendale di riferimento è SpazioSei s.r.l. (di seguito SpazioSei), azienda che produce e commercia abbigliamento per l'infanzia ed è stato svolto in collaborazione con Quix s.r.l. che rappresenta uno degli attuali fornitori di SpazioSei.

SpazioSei ha intrapreso un percorso di rinnovamento del proprio sistema informativo attraverso una serie di miglioramenti incrementali a partire da uno stato precedente, "ereditato" dal passato (Legacy System), rappresentato dall'attuale gestionale, al quale sono stati integrati nuovi applicativi. Integrare significa utilizzare il sistema ed i dati legacy nel più ampio contesto del business e della sua architettura informativa. L'integrazione nasconde (incapsula) il sistema legacy dietro interfacce consistenti con lo scopo di farlo cooperare con gli applicativi di ultima generazione introdotti.

Il progetto punta a favorire tale integrazione e per la sua realizzazione sono stati individuati i seguenti obiettivi (come mostrato in fig. 1):

1. Studio della procedura di estrazione e dati riguardanti l'anagrafica prodotti (ad es: stagioni, linee, articoli, colori, ecc.).
2. Definizione della tabella di mapping per l'importazione dei dati, dell'anagrafica prodotti, in una base di dati. Questa tabella dovrà descrivere la sorgente dati che potrà essere di natura diversa:
  - File a lunghezza prefissata;
  - File a separatore (csv);
  - Tabella di database.
3. Implementazione della procedura standard di importazione dati dell'anagrafica prodotti.

Requisito fondamentale di questo sistema è rendere la procedura di importazione flessibile. Un eventuale campo aggiuntivo dovrà prevedere solamente l'aggiunta del campo nella tabella di mapping senza dover modificare la procedura di importazione.

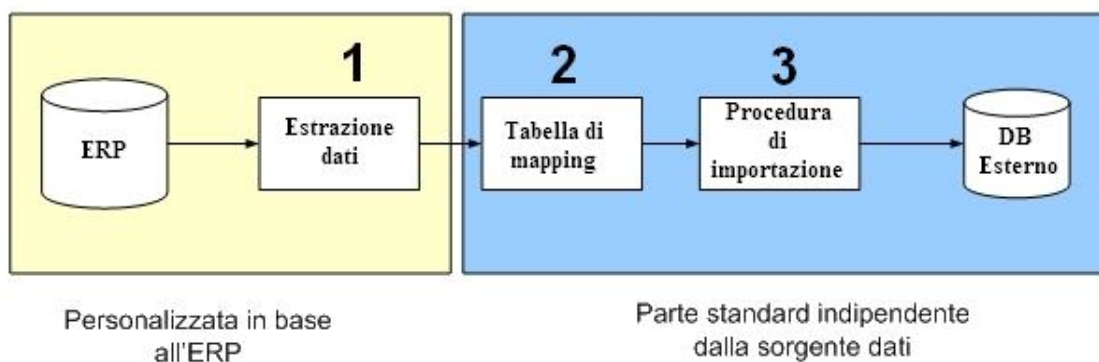


Fig. 1 Schema riassuntivo delle parti in gioco

La tesi si articola in cinque capitoli:

Nel Capitolo 2 vengono introdotti i concetti di azienda, informazione, sistema informativo, sistema informatico, ERP e Legacy System.

Nel Capitolo 3 viene trattato il Sistema Informativo di SpazioSei e la sua evoluzione. In particolare è analizzato l'ERP Legacy, e come sono state affrontate le esigenze di business.

Nel Capitolo 4, il paragrafo 4.1 è dedicato all'analisi della base di dati dell'ERP Legacy. Nel paragrafo 4.2 vi è l'analisi della base di dati realizzata da Quix s.r.l., su cui lavorano le applicazioni integrate all'ERP, che rappresenta la destinazione dei dati importati attraverso la procedura standard. I paragrafi 4.3, 4.4 e 4.5 descrivono le tre fasi di realizzazione del progetto di tesi (vedi Fig.1). Il paragrafo 4.6 riporta gli strumenti utilizzati.

Il Capitolo 5 delinea le conclusioni e gli sviluppi futuri.

## **2. IL SISTEMA INFORMATIVO AZIENDALE**

Un'azienda è un insieme di beni organizzati e coordinati dall'imprenditore o dal management per raggiungere obiettivi lucrativi cioè di conseguimento del reddito oppure non lucrativi ad esempio per scopi religiosi, sportivi sociali o culturali.

L'informazione è un bene utilizzato per controllare e pianificare le attività aziendali con efficacia, come le materie prime vengono trasformate dai sistemi di produzione così l'informazione è la materia prima del sistema informativo. Non bisogna confondere il termine informazione con il termine "dato", esiste infatti una differenza sostanziale. I dati rappresentano dei fatti della realtà in esame, ad esempio il fatto che un articolo abbia un certo prezzo sarà rappresentato da un numero (il dato) posto da qualche parte nel sistema informativo. Si deve osservare che il dato da solo non può portare alcuna informazione, ad esempio se ci viene comunicato il dato 2342 senza altre informazioni, esso potrebbe essere riferito a qualsiasi cosa all'interno dell'azienda: il prezzo di un articolo, il suo peso, il codice di un cliente. Il dato costituisce informazione solo se è in grado di modificare la conoscenza della realtà di chi lo riceve.

Il Sistema informativo Aziendale (SIA) è un insieme organizzato di procedure e risorse umane e materiali utilizzate per la raccolta, l'archiviazione, l'elaborazione e la comunicazione di informazioni necessarie ad una organizzazione per gestire sia le attività operative che di governo.

All'interno del sistema informativo vi è il sistema informatico che è il sottoinsieme del sistema informativo dedicato alla gestione automatica di informazioni e che fa evidentemente uso di materiali di tipo informatico (ad esempio elaboratori, periferiche varie, dischi, nastri).

Possiamo dire che ogni organizzazione al cui interno sia necessario produrre o scambiare informazioni, dispone in modo consapevole o inconscio, di un proprio sistema informativo.



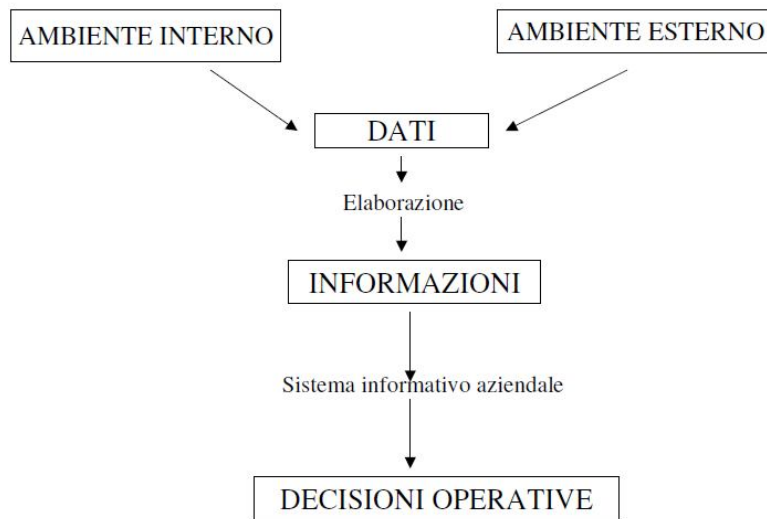


Fig.2.1 Schema di costruzione ed utilizzo delle informazioni aziendali

Vi sono varie parti che costituiscono un Sistema Informativo Aziendale:

- Patrimonio di dati
- Insieme di procedure per l'acquisizione e il trattamento dei dati e la produzione di informazioni
- Insieme di persone che sovrintendono tali procedure
- Insieme di mezzi e strumenti necessari al trattamento, trasferimento ed archiviazione di dati ed informazioni
- Insieme di principi generali di valori, di idee di fondo che caratterizzano il sistema e ne determinano il comportamento



Fig.2.2 elementi di un SIA

Lo scopo di un SIA è quello di raccogliere e classificare le informazioni producendo in tempo utile all'azienda le sintesi necessarie per i processi decisionali per poter così gestire e controllare l'attività aziendale nel suo complesso. Un SIA si dice informatizzato quando utilizza tecnologie informatiche per la gestione delle informazioni, i benefici che si traggono dal SIA informatizzato sono:

- Risparmio di tempo
- Sicurezza: elaborazione dati efficiente e sicura, back-up
- Efficienza: ottimizzare l'impiego di risorse
- Vantaggio competitivo, informazione utilizzata per nuovi scopi.

Le risorse materiali di tipo non informatico degli attuali sistemi informativi (es. archivi cartacei) si stanno sempre più riducendo al punto che i termini Sistema Informativo e Sistema informatico si possono spesso considerare sinonimi.

Gli obiettivi di un SIA sono:

- Produrre le informazioni destinate a supportare le decisioni dei soggetti aziendali
- Creare una struttura organizzativa che risponde alle esigenze aziendali
- Favorire la comunicazione tra i diversi soggetti aziendali ed i soggetti esterni all'azienda (clienti, fornitori, agenti)
- Velocizzare e facilitare il lavoro dei soggetti interni all'azienda
- Fornire analisi dettagliate sull'andamento aziendale

Il Sistema informatico è costituito da un insieme di apparecchiature in grado di ricevere dati in *input*, di operare su di essi fornendo in uscita *output* le informazioni necessarie per la gestione dell'azienda. Il sistema informatico cambia nel tempo con l'evolversi dell'azienda e delle tecnologie. L'insieme di questi strumenti elettronici e la gestione dei dati prende il nome di Electronic Data Processing (EDP).

## 2.1 ERP

L'elaborazione elettronica dei dati permette di realizzare un sistema di contabilità integrata che fornisce la possibilità di accedere alle informazioni a tutti i soggetti aziendali. Tale sistema integrato consente di ottenere sia documenti che rispondono agli obblighi contabili (fatture, note di accredito) sia di predisporre documenti non contabili (statistiche, report, grafici). La trasmissione dei dati è realizzata tramite una rete aziendale (sistema ERP - Enterprise Resource Planning) a cui sono collegate le unità di elaborazione delle diverse funzioni aziendali e consente l'accesso diretto alle informazioni a tutti gli utenti autorizzati.

Il sistema ERP presenta le seguenti caratteristiche:

- È **flessibile** in quanto consente di operare tutti i cambiamenti esterni o interni all'azienda
- È **modulare** esistono insiemi di funzionalità destinate ad uno specifico processo o funzione aziendale (es. modulo contabile per la gestione degli adempimenti amministrativi, produzione per la gestione delle informazioni sulla configurazione del prodotto ecc.)
- È **economico** cioè fornisce nei casi migliori un elevato ROI ottimizzando le operazioni di inserimento e consultazione dei dati
- È **gestionale** in quanto è possibile archiviare e rielaborare grandi quantità di dati condivisi in un'unica base di dati comune
- È **organizzativo** in quanto è facilitata la comunicazione tra le diverse funzioni aziendali

Il termine ERP viene comunemente utilizzato per applicazioni realizzate da vendor internazionali che operano in molteplici paesi ed hanno raggiunto un elevato grado di diffusione, in particolare nelle imprese di maggiori dimensioni. In Italia caratterizzata da PMI spesso più che di ERP si parla di gestionali.

## 2.2 Legacy System

La letteratura ha spesso fornito diverse definizioni di *legacy system*; dall'analisi del termine "legacy" ("qualcosa di valore ricevuto dal passato"), si può considerare la seguente definizione [1]:

*“ i sistemi software legacy sono stati sviluppati decenni fa e sono stati continuamente modificati per rispondere ai cambiamenti nei requisiti commerciali e nelle piattaforme di calcolo. La proliferazione di tali sistemi provoca grossi problemi per le grandi aziende che trovano costosa la manutenzione, ma pericolosa l'evoluzione”*

Di questa definizione, bisogna approfondire il significato dei termini "sviluppati decenni fa" e "costosa la manutenzione, ma pericolosa l'evoluzione". Il primo vuole indicare generalmente che il sistema che si prende in considerazione è già operativo, da almeno cinque anni. Il secondo indica che se il sistema non risponde più ai requisiti di business aziendali occorrerà sostituirlo, oppure se risponde a bisogni parziali come quelli del personale interno e si comporta in modo affidabile viene integrato con applicazioni che forniscono nuovi servizi ai soggetti esterni all'azienda.

### 2.2.1 Caratteristiche di un Legacy System

Le caratteristiche fondamentali di un *legacy system* sono:

È un sistema "mission-critical", cioè supporta il core delle operazioni aziendali interne ed è costantemente utilizzato quindi fondamentale per l'operatività dell'organizzazione.

Su di esso l'organizzazione ha pesantemente investito nel corso degli anni con personalizzazioni tagliate sulle esigenze aziendali del momento, e quindi non può essere semplicemente accantonato.

È di grosse dimensioni, centinaia di migliaia di files e linee di codice, distribuite su migliaia di programmi.

Il suo nucleo risale ad un decennio fa ed è quindi progettato secondo vecchie concezioni, che portano a problemi di scalabilità e flessibilità.

Basato su tecnologie non più allo stato dell'arte, scritto in linguaggi di vecchia generazione (RPG, Cobol), con DBMS obsoleti ed interfaccia utente a caratteri (Es. Terminale 5250);

Non è ben documentato (spesso non esiste proprio documentazione) ed è difficile da comprendere, poiché la documentazione non è aggiornata con le modifiche che via via sono state apportate, non esiste documentazione sulla sua evoluzione;

Il sistema è considerato come il repository di funzionalità aziendali non esplicitamente documentate che possono arrivare a gestire processi molto particolari (quali quelli esistenti nel settore tessile o del legno e arredo).

Spesso non è standard, a causa delle continue personalizzazioni ogni installazione è diversa dall'altra.

E' possibile considerare Legacy System la maggior parte delle applicazioni sviluppate negli anni '70 e '80. Purtroppo le caratteristiche sopra menzionate, possono riguardare anche applicazioni di recente realizzazione.

## 2.2.2 Tipologie di Legacy System

Come riporta [2] i sistemi e le applicazioni legacy, possono essere classificati come:

**1) Altamente Decomponibili:** ben strutturati e presentano alcune caratteristiche fondamentali:

- a) I componenti applicativi sono separabili in logica di presentazione, logica applicativa e logica d'accesso ai dati, cioè il software è composto di tre livelli logici;
- b) I moduli applicativi sono indipendenti tra loro;
- c) I moduli applicativi hanno interfacce ben definite con i servizi di database, quelli di presentazione e le altre applicazioni;

**2) Data decomponibili:** sono i cosiddetti "*semistrutturati*" con le seguenti caratteristiche fondamentali:

- a) I componenti applicativi sono separabili in due livelli: i servizi d'accesso ai dati e quelli di presentazione e logica applicativa fusi in un solo blocco;
- b) I moduli applicativi hanno interfacce ben definite verso le altre applicazioni. In questi sistemi è possibile accedere direttamente ai dati, ma non alla logica applicativa.

**3) Program decomponibili:** sono anch'essi semistrutturati con le seguenti caratteristiche fondamentali:

a) I componenti sono separabili in due livelli: i servizi di presentazione e quelli di accesso ai dati e logica applicativa fusi in un unico blocco;

b) I moduli applicativi hanno interfacce ben definite verso le altre applicazioni. In questi sistemi non è possibile accedere direttamente ai dati, ma necessario invocare delle funzioni predefinite (tipicamente una transazione). In questa categoria rientrano la maggior parte delle applicazioni legacy;

**4) Monolitici (non strutturati):** sono i sistemi in cui tutti i componenti appaiono come un unico blocco in cui tutti i livelli logici fusi insieme. Generalmente a questi sistemi si può accedere solo attraverso l'invocazione da interfaccia grafica.

Molti sistemi legacy hanno una architettura formata dalla combinazione di queste quattro. Dal punto di vista della facilità di trattamento, i Legacy System possono essere distinti in:

*Ostili:* sono quelli che non permettono la possibilità d'interfacciamento con l'esterno;

*Trattabili:* l'interfacciamento con altri sistemi è possibile con un certo sforzo di programmazione e tecnologie ad hoc;

*Amichevoli:* l'interfacciamento con altri sistemi è facilmente fattibile.

Risulta evidente che i sistemi del primo tipo sono amichevoli, quelli data/program decomponibili sono trattabili, mentre quelli dell'ultimo tipo sono ostili.

### 2.2.3 Trattamento dei Legacy System

Fino a qualche anno fa l'unica attività operativa legata in particolar modo ad un sistema Legacy era la manutenzione, sia evolutiva che correttiva. L'elevato costo della manutenzione per questi sistemi, ha però portato ultimamente sia le software house che gli stessi loro clienti, alla valutazione di possibili alternative per il trattamento dei sistemi Legacy:

**Esclusione:** non si sviluppa nessuna nuova funzione. È una soluzione non praticabile se il sistema contiene un alto valore per l'azienda.

**Sostituzione Netta:** si riscrive il sistema completamente da zero: "reengineering" del sistema.

**Migrazione graduale:** si opera una trasformazione del sistema in maniera graduale, migrando verso nuove tecnologie. In genere viene migrato un modulo alla volta.

**Integrazione:** si consolida il sistema nelle sue funzionalità e si effettua il "wrapping" con tecnologie ad hoc, integrandolo con applicativi che svolgono le nuove funzioni richieste.

Tra queste possibilità è possibile fare una prima distinzione a seconda se l'obiettivo sia la sostituzione con un nuovo sistema, o il mantenimento di gran parte del sistema adattato con nuovi servizi e nuove tecnologie. Questo, ovviamente dipende dal valore che l'azienda attribuisce al sistema, in termini di servizi forniti e criticità.

### **3. IL SISTEMA INFORMATIVO DI SPAZIOSEI**

L'azienda è nata nel 1992 e da allora ha avuto una crescita costante che l'ha portata al top nel settore dell'abbigliamento infantile. Per questo da una linea di abbigliamento si è giunti a contarne quattro, due aziendali e due in licenza, con sviluppo di linee che abbracciano diverse fasce di età e merceologie per un totale di dieci collezioni a stagione. L'azienda è situata presso uno dei distretti produttivi con la più spiccata vocazione a realizzare prodotti di alto contenuto creativo e di eccellente contenuto qualitativo. A Carpi, celebre cittadina dell'Emilia Romagna, ha, infatti, sede e operatività l'azienda che da sempre è molto attenta alla formazione dei propri dipendenti che attualmente sono circa un centinaio. La politica aziendale, sin dagli esordi, ha risposto con coerenza alle istanze del mercato, ponendo l'accento sull'unicità del prodotto: design contemporaneo, materiali della migliore qualità, lavorazioni di prim'ordine. E non da ultimo, attenzione all'ambiente e alla sostenibilità attraverso l'alimentazione a energia solare e il riciclo di materiali riutilizzabili.

La crescita economica aziendale è stata supportata da un'evoluzione del Sistema Informativo che nel tempo si è dovuto adattare alle sempre nuove esigenze di business.

Come introdotto in [3] :

“Lo sviluppo delle potenzialità offerte dalla recente evoluzione tecnologica ha sancito il passaggio all'era dell'integrazione tra sistemi, che si è riversata nel concetto di elaborazione ubiqua resa possibile, a cavallo del nuovo millennio, dalle possibilità messe a disposizione dalle nuove tecnologie di comunicazione. L'avvento delle tecnologie della comunicazione, verificatosi nel passaggio da IT ad Information and Communication Technology (ICT), ha indotto a ragionare in termini di sistema informativo integrato, ma anche a riflettere sugli aspetti di processo, sugli aspetti organizzativi e sull'importanza che gli strumenti aggiungano effettivamente una marcia ai processi di creazione del valore. Quest'ultima evoluzione o passaggio è quello che viene definito “era dell'elaborazione ubiqua”, la quale è una conseguenza diretta della diffusione di Internet e delle tecnologie digitali di comunicazione. ”

Possiamo affermare che l'azienda è consapevole dell'importanza che l'innovazione informatica ricopre, è importante pertanto poter disporre di strumenti all'avanguardia per conseguire al meglio gli obiettivi economici prefissati. Nei prossimi paragrafi andremo ad



analizzare in dettaglio il sistema informativo e le motivazioni che l'hanno spinto nell'adozione di un approccio orientato all'integrazione dell'ERP.

### 3.1 L'ERP

Il gestionale aziendale è sviluppato su una piattaforma proprietaria IBM iSeries 5 [4] comunemente chiamata As400 [5], sistema operativo Operating System/400 (OS/400) che poggia su una base di dati DB2. L'ERP presenta due moduli uno applicativo contabile finanziario ed uno applicativo per la produzione. Il gestionale ricopre un buon numero di funzioni, è un sistema progettato e sviluppato ad hoc, cioè costruito in modo specifico sulle esigenze dell'azienda, è un gestionale di tipo Verticale<sup>1</sup> tipologia diffusa nelle aziende di abbigliamento. L'ERP di SpazioSei è a tutti gli effetti un Legacy System, infatti la tecnologia di sviluppo del gestionale risale a più di 30 anni fa, ed ogni funzione che implementa è scritta in linguaggio RPG (linguaggio sviluppato nella fine degli anni 70). L'interfaccia utente è un terminale a caratteri 5250. L'accesso al gestionale è disponibile anche attraverso un'interfaccia grafica iSeriesNavigator che permette di connettere "velocemente" il terminale al server e gestire le code di stampa, ma che risulta ostile ad un utente medio.

Da classico sistema Legacy, l'ERP soddisfa requisiti di base legati all'esterno, quali ad esempio, requisiti legali, civilistici e fiscali, di relazione con banche, soggetti stabili che pongono le loro condizioni e dettami per poter instaurare un rapporto di affari. Fotografa situazioni molto specifiche cristallizzandole e limitando le possibilità di adeguamento dei processi alle sollecitazioni esterne o anche al rinnovamento dell'organizzazione interna.

A differenza di quanto detto nel precedente paragrafo (Cfr. Par. 2.1) non rispecchia appieno le caratteristiche di un ERP, infatti, questo tipo di sistema, realizzato in modo artigianale,

---

<sup>1</sup> Si tratta di applicazioni che presentano una significativa componente specializzata per supportare i processi di alcuni settori industriali specifici. In alcuni di questi settori rappresentano il sistema più diffuso in quanto generalmente ben aderenti alle richieste più comuni delle imprese. Sono realizzati da software house nazionali o locali, che hanno sviluppato nel tempo metodologie consolidate per l'implementazione, permettendo così di contenere i rischi di adozione da parte dei clienti; in genere, l'implementazione di un verticale di settore risulta più agevole per l'impresa e comporta minori probabilità di superamento dei limiti di tempo e di budget definiti.

ritagliato sulle peculiari esigenze dell'utente, presenta alcune lacune come la mancanza di una adeguata copertura dei processi di pianificazione e controllo; le fasi di pianificazione, budgeting e verifica delle prestazioni sono infatti spesso realizzate tramite reportistica, statistiche a posteriori che, se confrontate con i moderni strumenti software oggi a disposizione, risultano assolutamente inadeguati. Le funzioni progettate sono organizzate in maniera piuttosto rigida, per uffici, per reparti, per funzioni appunto e non sono in grado di dare una versione integrata dell'organizzazione nel suo complesso, risultando obsolete rispetto alle necessità di crescente integrazione delle attività di business, favorendo una proliferazione di ridondanze, inconsistenze ed incompletezza dei dati. Questo sistema non è adatto a fare analisi per il supporto alle decisioni, le query di analisi in questo ambiente sono molto complesse da formulare, hanno una alta complessità di calcolo con Join su molte tabelle, non sono interattive. Una complessità aggiuntiva è data anche dalla separazione degli archivi: esiste un sistema OLTP dove le informazioni amministrative sono separate da quelle di produzione.

Nella classificazione dei Legacy System il gestionale in questione si può classificare come program decomponibile (Cfr. Par. 2.2.2). La base di dati presenta migliaia di file ognuno contenente altrettanti record. La struttura è complessa, le informazioni sono contenute in molte tabelle spesso annidate ("tabelle di tabelle") che aggiungono ridondanze e complessità alla logica di interrogazione. Un esempio concreto lo abbiamo avuto durante l'integrazione con un applicativo di Business Intelligence. La valorizzazione delle righe d'ordine (utile per una analisi del venduto durante la campagna vendite) non veniva salvata nella base di dati dell'ERP ma calcolata ogni volta che un utente produceva una query, ad esempio semplicemente lanciando una stampa. Il risultato faceva sì che risultasse difficile, per la Business Intelligence, reperire un "dato certificato" che ne permettesse analisi più puntuali e rapide. Vi era così anche l'impossibilità di avere uno storico dei dati perdendo un'informazione importante per il management. Ricreare la logica del gestionale sulla Business Intelligence sarebbe stato troppo complesso (interrogazioni su tante tabelle) ed avrebbe aggiunto delle criticità come ridondanze, disallineamento dei dati ed errori di calcolo, oltre a costi di analisi maggiori in quanto i due applicativi provenivano da fornitori diversi. La soluzione è stata quella di aggiungere quindi dei campi nella base di dati dell'ERP dove poter salvare questi dati. L'operazione è stata comunque complessa ed ha richiesto notevole dispendio di tempo e denaro, la difficoltà stava nel ricostruire il flusso di dati che le varie

procedure di calcolo andavano a considerare, aggiungere i campi richiesti in ogni tabella considerata (ridondanza), riscrivere le procedure ed aggiornare i dati esistenti (storico).

Come vedremo nel prossimo paragrafo la società ha deciso di mantenere il proprio gestionale in quanto copre i processi core, e vi è stato nel corso degli anni un grande investimento economico per la sua evoluzione. Dall'altra parte, come sottolineato precedentemente, SpazioSei ha avviato una profonda riorganizzazione dei propri processi integrando il gestionale ad applicazioni atte a supportare le nuove esigenze di business.

### 3.2 Nuove esigenze di business

SpazioSei è un perfetto esempio di *Extended Enterprise*, i cui processi travalicano i confini aziendali e l'organizzazione interna si *mixa* con quella esterna.

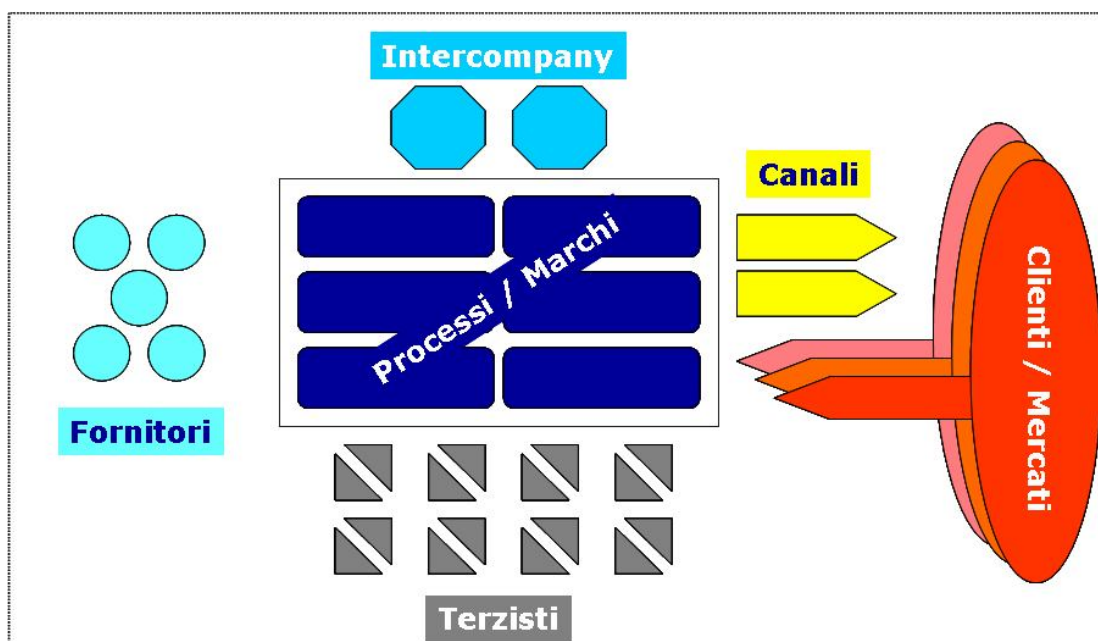


Fig.3.1 attori coinvolti nel business aziendale

In ambito ICT si sta attuando un cambio di paradigma, che permette senza discontinuità, il passaggio dall'attuale sistema Legacy ad un nuovo modello ICT in ottica SOA – *Service Oriented Architecture*.

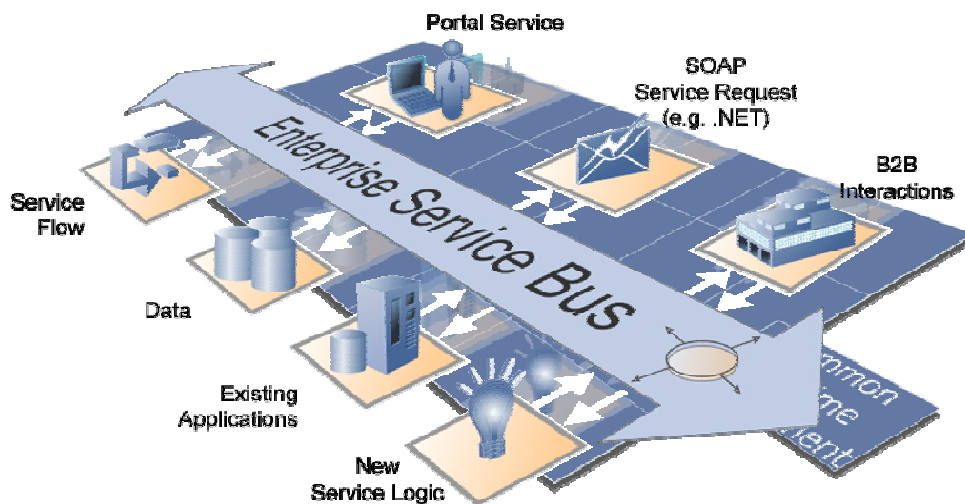


Fig.3.2 esempio di struttura SOA

Per fare ciò, si sta costruendo il nuovo Sistema Informativo costituito da un mix di tecnologie eterogenee in ottica “best of bread” che consiste nell’acquisto di specifici strumenti per ogni area funzionale in grado di consentire lo sviluppo di strati software per l’integrazione. Sono stati inseriti all’interno di un portale: un software di Business Intelligence di supporto al management, un dipartimentale per il Retail<sup>2</sup> e come supporto alla rete vendita un applicativo di inserimento ordini. L’approccio di questi anni è stato pertanto modulare, in questo modo si sono potuti sfruttare al meglio gli investimenti ICT già effettuati ed inserire/sostituire le nuove componenti necessarie ad abilitare i servizi di business via via richiesti. In futuro si andranno ad implementare un sistema di gestione documentale ed un CRM.

Si sta quindi costruendo l’Impresa Digitale in quanto potente Strumento:

- organizzativo = integrazione dei processi
- informativo = integrazione delle informazioni
- collaborativo = integrazione delle persone
- e di governo dell’interno SIA (eterogeneo) = integrazione delle applicazioni/servizi

Per le proprie caratteristiche, tale *Strumento* risulta propedeutico sia alla riorganizzazione aziendale, che ad un cambio futuro del sistema ERP, permettendo un passaggio più facile e graduale, dal vecchio al nuovo, e lasciando la possibilità di farli coesistere nella percentuale

<sup>2</sup> Attività di distribuzione commerciale al dettaglio.

voluta. Questo concetto si può tradurre in termini di “scalabilità” e “flessibilità” del sistema informativo.

Grazie alla **Digital Enterprise** ed alle specifiche tecnologie da adottare congiuntamente ad essa, i servizi vengono distribuiti all’utenza, interna ed esterna, in funzione delle autorizzazioni legate al proprio ruolo: proprietà, direzione, dipendenti, agenti, clienti, prospect, fornitori, terzisti, ecc...



Fig.3.3 Servizi di front-end e di back-end

Le nuove applicazioni utilizzano una base di dati MySQL contenente informazioni importate direttamente dall’ERP. Tali applicazioni sono state interfacciate al data base del sistema Legacy tramite connettori realizzati ad hoc utilizzando procedure ETL anche complesse che presentano costi di gestione da non sottovalutare. Allo stato attuale l’importazione di nuovi dati dall’ERP implica il coinvolgimento di un programmatore o nei casi più complessi anche di un analista per modificare la procedura di importazione dell’ETL; per fare questo occorre sapere dove e come recuperare i dati, entrando nella logica del gestionale (non sempre semplice). In quest’ottica è nata l’idea di implementare la procedura standard di importazione dati da sorgenti differenti all’interno di una base di dati MySQL. La base di dati MySQL è attualmente utilizzata come DataWarehouse, in quanto adatta ad applicazioni sviluppate per il web e per il DataMining sulla quale i “nuovi applicativi” stanno già lavorando. Per lo sviluppo del progetto è stata utilizzata una copia ridotta<sup>3</sup> di questa base di dati che andremo ad analizzare nel prossimo capitolo.

<sup>3</sup> Si è deciso di operare su una copia della base di dati, attualmente in produzione, per motivi di sicurezza, di complessità e tempi di realizzazione.

## 3.3 Da DB2 a MySQL

Analizzeremo brevemente le basi di dati DB2 e MySQL mostrandone vantaggi e svantaggi.

### 3.3.1 Panoramica generale DB2

Il DBMS DB2 [6] e [7] appartiene alla storica famiglia della IBM che pone le sue fondamenta in SYSTEM R uno dei primi prototipi dei DBMS relazionali mai sviluppati, risalente ai primi anni settanta. La prima versione di DB2 risale alla prima metà degli anni '80 e da allora questo prodotto IBM è sempre stato all'apice del mercato mondiale dei DBMS, insieme ad ORACLE.

Il successo che ha sempre caratterizzato DB2 nasce da una collezione di funzionalità avanzate, quali:

- Supporto alla gestione di tipi di dati non tradizionali
- La possibilità di creare dei tipi di dato e delle funzioni che l'utente può utilizzare semplicemente nell'SQL
- Supporto nativo alle applicazioni web
- Funzionalità evolute di Business Intelligence
- Supporto per parallelismi sia a memoria condivisa sia a basi di dati memorizzate su calcolatori basate su multiprocessori simmetrici (SMP), sia a parallelismi a memorie separate nei quali una base di dati viene partizionata tra diversi calcolatori connessi in rete
- Naturalmente, la componente server di DB2 è disponibile per diverse piattaforme sia software che hardware ed oltre ad essere ovviamente disponibile per gli ambienti Unix e Linux è anche disponibile per gli ambienti Windows.

Un'ulteriore caratteristica interessante di DB2 è che i sistemi DB2 possono partecipare a federazioni eterogenee e distribuite di basi di dati, usando un protocollo chiamato DRDA (Distributed Relational Database Architecture).

Tuttavia visto il grandissimo successo riscosso da DBMS gratuiti come MySQL visto inoltre l'andamento del mercato, l'IBM ha rilasciato DB2 Universal Database Express-C, dove la "C" sta per community edition; questa particolare versione poggia sullo stesso motore delle versioni enterprise e condivide le stesse interfacce di programmazione e gli stessi limiti di

sistema dell'edizione Express a pagamento. [...] DB2 Express-C è indirizzato soprattutto agli sviluppatori che necessitano di un DBMS compatto da inglobare all'interno delle proprie applicazioni scritte in C/C++, Java, MS.NET, PHP e altri linguaggi. La licenza che accompagna il software, pur non essendo open source, permette agli utilizzatori di “scaricare, sviluppare, installare, testare, eseguire, inglobare e ridistribuire” il programma e i suoi componenti. E' possibile reperire il DB2 Universal Database Express-C al sito:  
<http://db2express.com/it/> .

### 3.3.2 Panoramica e storia di MySQL

MySQL è uno dei più diffusi sistemi DBMS open source, sviluppato, distribuito e supportato da MySQL AB, un'organizzazione fondata dagli stessi, primi sviluppatori di MySQL. A riprova della sua grossa diffusione, si consideri che, oltre alla vastissima utenza, MySQL è attualmente in grado di operare su un parco di oltre 20 piattaforme, tra le quali annoveriamo Linux, Windows, OS/X, HP-UX, AIX, Netware.

### 3.3.3 Caratteristiche di MySQL

MySQL può essere inquadrato nell'ambito dei database di tipo relazionale. Per database relazionale si intende un database nel quale i dati immagazzinati sono memorizzati sotto forma di tabelle, messe poi in relazione tra loro; questa tecnica organizzativa permette di ottimizzare la gestione dei dati, ma soprattutto di ridurre lo spazio di memoria impegnato.

MySQL è quindi un software fornitore di un sistema in grado di gestire i database relazionali. In questa accezione, possiamo indicare MySQL non propriamente come DBMS, bensì più precisamente come RDBMS, ovvero Relational DataBase Management System (Sistema di gestione di database relazionali). Lo stesso DB2, di IBM, del quale si è parlato nel precedente paragrafo è un RDBMS.

MySQL utilizza il linguaggio SQL (Structured Query Language) per permettere, principalmente, l'interrogazione dei database che gestisce. Oltre a questa funzione principale, questo particolare linguaggio permette di effettuare tutte le operazioni necessarie alla creazione delle tabelle di un database; di definire i vincoli intrarelazionali (ossia vincoli vigenti all'interno di un'unica relazione) ed interrelazionali (ossia vigenti tra due o più relazioni); di inserire, cancellare e modificare i dati contenuti nel database.

MySQL è un prodotto open source, distribuito con licenza GPL. Può dunque essere liberamente prelevato attraverso Internet al [8].

### 3.3.4 Vantaggi di MySQL 5.0

La scelta di importare i dati da DB2 a MySQL è stata fatta in quanto la priorità assoluta, per chi non vuole perdere tempo, è assicurarsi di immagazzinare i propri dati in formati standard, in quanto solo questi assicurano una permanenza di disponibilità nel futuro e una maggiore portabilità verso altre basi di dati. Di seguito riportiamo alcuni punti forza di MySQL [9]:

- Gratis per l'utilizzo come Open Source
- Eccezionale diffusione, soprattutto per le applicazioni web e su server Linux
- Un ottimo e completo SQL utilizzabile direttamente o con i più diffusi linguaggi di programmazione
- Disponibile anche con una licenza commerciale e supporto
- Leggero e di poco impatto sui server su cui viene installato
- Semplice nell'utilizzo, nella configurazione, nell'amministrazione
- Ottime prestazioni
- Consente l'utilizzo di differenti Storage Engine tra cui scegliere: operazioni dirette in memoria per le massime prestazioni, gestione completa delle transazioni o HA clustering

Come riportato dal sito ufficiale [8], MySQL riduce il costo totale di proprietà di software database riducendo i costi di licenza di oltre il 90 per cento e i downtime di sistema del 60 per cento. Al tempo stesso, riduce le spese per l'hardware del 70 per cento e i costi di amministrazione, engineering e supporto di oltre il 50 per cento.



## 4. PROCESSO DI IMPORTAZIONE

Il processo è articolato in varie fasi (vedi fig. 4.1). Vi è un' estrazione dalla base di dati dell'ERP AS400, delle anagrafiche interessate. I dati estratti rappresentano una minima parte delle informazioni contenute in questa base di dati. L'estrazione dei dati dall'ERP è personalizzata in base alla base di dati del gestionale (bisogna conoscere il tracciato record). I dati estratti vengono salvati in file .CSV (punto 1 fig. 4.1), vengono creati un file contenente i dati delle stagioni, un file contenente i dati delle linee ed un file per gli articoli. L'estrazione opera una trasformazione dei dati per permettere alla procedura di importazione Standard di importarli nella base di dati di arrivo MySQL. Tra i file .CSV e la procedura di importazione standard vi è una tabella di mapping che fornisce informazioni utili alla procedura Java per effettuare l'importazione (vedi punto 2 fig. 4.1). La procedura in Java utilizzerà la tabella di mapping per importare i dati nella base di dati MySQL (fig.4.1 punto 3). Nei prossimi paragrafi verranno approfondite le fasi descritte.

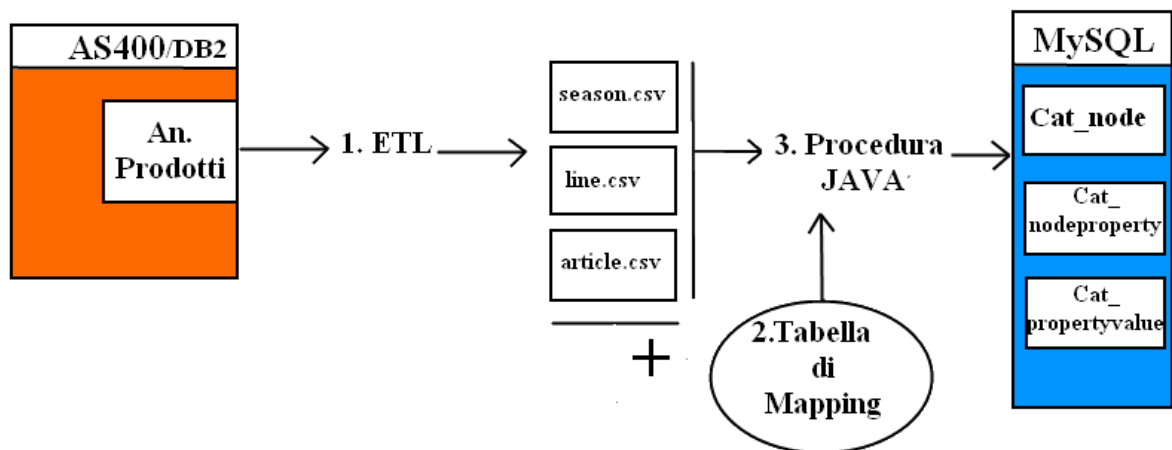


Fig. 4.1 flusso delle operazioni

## 4.1 Base di dati Legacy

In questa parte analizzeremo la struttura delle anagrafiche dei prodotti<sup>4</sup> ed alcuni file di interesse contenuti nella base di dati dell'ERP Legacy. Questa analisi è doverosa in quanto permette di individuare i dati da estrarre tramite l'ETL.

### 4.1.1 Anagrafica prodotti

L'anagrafica dei prodotti è importata direttamente dall'ERP aziendale. I prodotti sono organizzati in una gerarchia (fig. 4.2) caratterizzata dalla stagione e dalla linea. Tramite l'anagrafica prodotti è possibile risalire ad un determinato articolo in una determinata linea per una determinata stagione. La struttura anagrafica dei prodotti è paragonabile a quella di un albero dove alla radice ci sono dei nodi di tipo stagione e sotto dei nodi di tipo linea che contengono gli articoli. Ogni tipo di nodo (stagione, linea e prodotto) è caratterizzato da una serie di proprietà.

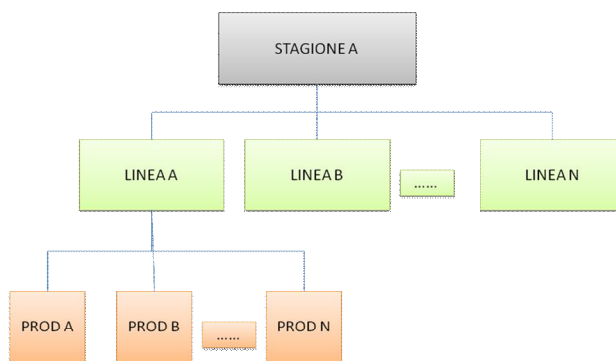


Fig. 4.2 Struttura anagrafiche prodotti

Di seguito vengono mostrati i campi che compongono i nodi che descrivono l'anagrafica Prodotto: Stagione, Linea e Prodotto.

Nome	Tipo	Valore	Descrizione
StatoRec	Testo	Unico	Identifica se la stagione è attiva o cancellata
Id	Testo	Unico	Identifica univocamente la stagione (coincide con il codice)
Code	Testo	Unico	Codice della stagione
Name	Testo	Unico	Descrizione della linea

Stagione

<sup>4</sup> Il termine "prodotto" nel mondo dell'abbigliamento di solito può essere sostituito dai termini "articolo" o "modello".

Nome	Tipo	Valore	Descrizione
StatoRec	Testo	Unico	Identifica se la stagione è attiva o cancellata
Id	Testo	Unico	Identifica univocamente la linea(codice stagione + linea)
Code	Testo	Unico	Codice della stagione di riferimento
Name	Testo	Unico	Descrizione della stagione

Linea

Nome	Tipo	Valore	Descrizione
StatoRec	Testo	Unico	Identifica se il prodotto è attivo o cancellato
Id	Testo	Unico	Identifica univocamente il prodotto(codice stagione + linea + prodotto)
Code	Testo	Unico	Codice della linea di riferimento (id della linea)
Name	Testo	Unico	Descrizione della stagione
SizeGroup	Testo	Unico, da lista	Segnataglie
Palette	Testo	Unico	Cartella Colori
Composition	Testo	Lista	Composizione tessuti
Color	Testo	Lista	Colori

Prodotto<sup>5</sup>

#### 4.1.2 Descrizione base di dati ERP

La base di dati dell'ERP è basata su "archivi", noti come "file". Gli archivi risiedono all'interno di librerie e sono messi a disposizione dal sistema di archiviazione (File System) del Sistema Operativo e dal linguaggio di programmazione RPG.

Ogni file è formato da una sequenza di record (registrazioni) su cui idealmente "scorre" un cursore del file. Il record corrente sarà quello dove si trova il cursore. Ogni record è strutturato in campi ciascuno dei quali ha un nome ed un tipo. Riassumendo, ogni dato giace quindi in un campo, che fa parte di un record, che fa parte di un file.

Un eventuale nuovo record si può aggiungere solo quando il cursore si trova subito dopo l'ultimo, cioè nella posizione *eof*<sup>6</sup>, non può essere inserito tra due record. Non esiste una

<sup>5</sup> Un prodotto nella realtà ha molte più proprietà, si è deciso di mostrare solamente quelle utilizzate per il progetto.

operazione specifica di cancellazione di un record (sarebbe troppo onerosa), ma vi è una “cancellazione logica” che setta lo stato record da “A” attivo a “C” cancellato (marca di cancellazione).

Il termine “dato” identifica il contenuto di una singola informazione (campo) su di un soggetto. Tutte le informazioni su di un soggetto sono raccolte in una descrizione (record) rispettando uno schema fisso (tracciato). Le descrizioni di più soggetti radunate insieme formano a loro volta un archivio (file).

Un esempio di quanto detto nel caso di SpazioSei, si ha per l’anagrafica Articoli (lo stesso vale per le Stagioni e per le Linee), dove le informazioni sono contenute in un gruppo (file) di schede descrittive (record) con una griglia (tracciato) di spazi predefiniti (campi) per riportare informazioni standardizzate (dati nei campi) come il codice articolo, la sua descrizione, il codice listino ecc.

E’ interessante analizzare la struttura dei file DSVSV e DSVLI che contengono i dati di Stagione e Linea. Questi sono definiti “membri” e sono contenuti in un file chiamato DSVAR che contiene quindi l’insieme dei membri. DSVAR risulta cioè composto da più tronconi che contengono gruppi di record diversi (Stagione e Linea hanno record differenti). I file risiedono all’interno di contenitori chiamati librerie, simili agli indirizzari di un pc, che a differenza di quanto succede in quest’ultimi, non possono contenere altre librerie, i file di una libreria sono sincronizzati tra loro.

File: DSVAR00F		Formato record: DSVAR		TRACCIATO RECORD TABELLE VARIE				
Nome Campo	Pos. Iniz.	Pos. Fin.	Lungh. Campo	Tipo	Num. Cifre	Num. Dec.	Nome Campo	Descrizione campo
SREC03	1	1	1	A			SREC03	Stato record
KEYX03	2	13	12	A			KEYX03	Chiave
DATI03	14	100	87	A			DATI03	Dati
=====								
Lunghezza Record:	100	Nr.Campi del file:		3				

Fig. 4.3 Tracciato record DSVAR

Analizzando il tracciato record in fig. 3.3 si deduce che i file contenuti in DSVAR hanno tutti la seguente struttura; il primo campo in posizione 1 rappresenta lo stato record, e la chiave è

<sup>6</sup> End Of File, fine file.

formata dai campi contenuti nelle 12 posizioni successive, per i dati vi è a disposizione uno spazio di 87 posizioni.

Per chiarire meglio andiamo ad analizzare il tracciato record del file DSVSV fig. 4.4 che è contenuto in DSVAR e riporta le informazioni relative alla stagione:

File: DSVSV		Formato record: DSVARSV		TRACCIATO RECORD TABELLA STAGIONI DI VENDITA					
Nome Campo	Pos. Iniz.	Pos. Fin.	Lungh. Campo	Tipo	Num. Cifre	Num. Dec.	Nome Campo	Descrizione campo	
TASV01	1	1	1	A			TASV01	STATO RECORD	
TASV02	2	3	2	A			TASV02	TIPO RECORD	
TASV03	4	11	8	A			TASV03	FILLER01	
TASV04	12	13	2	A			TASV04	CODICE STAGIONE	
TASV05	14	33	20	A			TASV05	DESCR. STAGIONE	
TASV06	34	34	1	S	1		TASV06	0=CONT 1=PRIM/EST 2=AUTUN/INV	
TASV23	35	35	1	A			TASV23	1=TRIANGOLAZIONE EURO	
TASV24	36	43	8	S	8		TASV24	DATA FINE PRODUZIONE	
TASV07	44	45	2	B	4		TASV07	ANNO DI COMPETENZA	
TASV25	46	46	1	A			TASV25	1=BLOCCO INS.ORDINI CLI.	
TASV08	47	48	2	A			TASV08	FILLER	
TASV09	49	50	2	S	2		TASV09	ANNO	
TASV10	51	52	2	S	2		TASV10	MESE	
TASV11	53	54	2	S	2		TASV11	GIORNO	
TASV12	55	58	4	A			TASV12	FILLER	
TASV15	59	62	4	A			TASV15	FILLER	
TASV14	63	68	6	S	6		TASV14	DATA INIZIO GESTIONE MATER.	
TASV16	69	74	6	A			TASV16	FILLER	
TASV17	75	78	4	P	6		TASV17	2* DATA SPEZZ.COMM.	
TASV18	79	82	4	P	6		TASV18	3* DATA SPEZZ.COMM.	
TASV19	83	86	4	P	6		TASV19	4* DATA SPEZZ.COMM.	
TASV20	87	90	4	P	6		TASV20	5* DATA SPEZZ.COMM.	
TASV21	91	94	4	P	6		TASV21	6* DATA SPEZZ.COMM.	
TASV22	95	98	4	P	6		TASV22	7* DATA SPEZZ.COMM.	

=====

Lunghezza Record: 98    Nr.Campi del file: 24

=====

Fig. 4.4 Tracciato record DSVSV

Si può notare che nella posizione 1 vi è lo stato record ('A' Attivo, 'C' Cancellato') la chiave di una stagione è formata dai campi contenuti nelle 12 posizioni successive, che presentano campi come Tipo Record che serve per individuare il file DSVSV (contiene la stringa "SV") in DSVAR, un campo Filler e il Codice Stagione. Nelle successive posizioni sono contenute le proprietà della stagione come descrizione data fine produzione ecc. Questi campi partono dalla posizione 14 ed arrivano alla 98. E' rispecchiata quindi la struttura di DSVAR (1 stato record, 12 chiave e 85 dati) con una "rimanenza" di 2. Ad esempio per chiarire si deve pensare alla posizione 14 del campo DATI03 di DSVAR nel caso del file Stagione DSVSV vi è la descrizione della stagione contenuta nel campo TASV05 che arriva fino alla posizione 33. Altro fattore importante è che questo file identifica solamente l'entità stagione ma non fornisce alcun parametro di relazione con le linee e gli articoli, quindi non è possibile tramite questo file risalire né alle linee né agli articoli appartenenti ad una stagione.

Il file DSVLI è analogo (vedi fig.4.5):

File: DSVLI		Formato record: DSVLI0		TRACCIATO RECORD TABELLA LINEE DI VENDITA TRK=BLANK					
Nome Campo	Pos. Iniz.	Pos. Fin.	Lungh. Campo	Tipo	Num. Cifre	Num. Dec.	Nome Campo	Descrizione campo	
TALI01	1	1	1	A			TALI01	STATO RECORD	
TALI02	2	3	2	A			TALI02	TIPO RECORD	
TALI03	4	4	1	A			TALI03	TIPO RECORD	
TALI04	5	12	8	A			TALI04	FILLER01	
TALI05	13	13	1	A			TALI05	CODICE LINEA	
TALI06	14	33	20	A			TALI06	DESCRIZIONE	
TALI13	34	34	1	A			TALI13	I=INGROSSO D=DETTAGLIO	
TALI01	35	43	9	A			TALI01	FILLER	
TALI09	44	47	4	S	4	2	TALI09	% RICARICO X CALC. LISTINI	
TALI10	48	51	4	S	4	3	TALI10	DIVISORE X CALCOLO LISTINI	
TALI11	52	52	1	A			TALI11	CODICE LISTINO	
TALI12	53	53	1	A			TALI12	LINEA CORRISPOND.X ANALIT.2	
TALI02	54	100	47	A			TALI02	FILLER	
Lunghezza Record: 100		Nr.Campi del file: 13							

Fig. 4.5 Tracciato record DSVLI

In posizione 1 ho lo stato record, dalla 2 alla 13 la chiave con tipo record TALI02 e TALI03, Filler e codice linea, poi dalla posizione 14 alla 100 le varie proprietà. Anche in questo caso non è possibile accedendo a questo file risalire né alla stagione di cui una linea fa parte e né agli articoli contenuti in una linea.

Alla luce di quanto detto DSVAR si può rappresentare come una tabella con una struttura ben articolata (vedi fig. 4.6).

		DSVAR																					
		SREC03	KEYX03											DATI03									
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	33	34	35	...	100
DSVSV	TASV01	TASV02	TASV03						TASV04		TASV05			TASV06	TASV23	...							
DSVLI	TALI01	TALI02	TALI03	TALI04						TALI05		TALI06			TALI13								

Fig. 4.6 Esempio struttura DSVAR

Le righe rappresentano i vari file contenuti in DSVAR, bisogna precisare che DSVAR contiene centinaia di file strutturati come descritto nei precedenti paragrafi. La limitazione di questa struttura sta nel fatto che se ad esempio per la tabella delle linee DSVLI vogliamo aggiungere una proprietà ma si sono sfruttate tutte le 100 posizioni disponibili di DSVAR, si deve aggiungere un file es. DSVLI1 che riprenderà la struttura stato record e chiave di DSVLI dove dalla posizione 14 ci saranno le nuove proprietà. Questo crea una duplicazione dei dati, infatti per avere tutte le proprietà di una linea si dovrà accedere con la stessa chiave sia a DSVLI che a DSVLI1.

L'anagrafica articoli (o anagrafica prodotti) è contenuta nel file DBANA che risiede in una libreria diversa da DSVAR. **DBANA mette in relazione una determinata linea ad una determinata stagione con un determinato articolo.** La chiave di un articolo è data dal codice stagione, dal codice della linea più il codice prodotto (che è formato dalla concatenazione dei campi CODICE MODELLO, CODICE ARTICOLO E MODIFICA, Vedi fig. 4.7) . E' importante sottolineare che **tramite questo file è possibile risalire alle linee appartenenti ad una stagione (CODICE STAGIONE più CODICE LINEA),** infatti osservando il tracciato record si possono riconoscere le chiavi di stagione e linea descritte precedentemente. Questo rappresenta un vincolo importante perchè **solamente durante la creazione dell'anagrafica di un prodotto viene associata una linea ad una stagione, non esistono pertanto stagioni e linee che non contengono articoli.**

TRACCIATO RECORD								
File:		ANAGRAFICA MODELLI/ARTICOLI						
DBANA								
Nome Campo	Pos. Iniz.	Pos. Fin.	Lungh. Campo	Tipo	Num. Cifre	Num. Dec.	Nome Campo	Descrizione campo
SREC6F	1	1	1	A			SREC6F	STATO RECORD
CSTA6F	2	3	2	A			CSTA6F	CODICE STAGIONE
CLIN6F	4	4	1	A			CLIN6F	CODICE LINEA
CMOD6F	5	8	4	A			CMOD6F	CODICE MODELLO
CART6F	9	12	4	A			CART6F	CODICE ARTICOLO
MODI6F	13	14	2	A			MODI6F	MODIFICA
QQ016F	15	16	2	A			QQ016F	FILLER
DES16F	17	36	20	A			DES16F	DESCRIZIONE MOD/ART
FL016F	37	37	1	S	1		FL016F	DB CREATA AUTOMATICAMENTE
CSET6F	38	38	1	A			CSET6F	CODICE SEGNATAGLIE
CE016F	39	40	2	P	3		CE016F	COD.ETICHETTA COLLO
CCAP6F	41	42	2	P	3		CCAP6F	CODICE CARTELLINO
CCOP6F	43	44	2	P	3		CCOP6F	CODICE CONFEZIONE
LISL6F	45	46	2	A			LISL6F	LISTINO LOGISTICA
TA016F	47	48	2	S	2		TA016F	VENDIBILITA' TAGLIE : DA ..
TA026F	49	50	2	S	2		TA026F	VENDIBILITA' TAGLIE : A ..
PPRP6F	51	52	2	P	3	1	PPRP6F	% PROVVISORIE PARTICOLARE
CIVA6F	53	54	2	A			CIVA6F	CODICE IVA
CFOP6F	55	57	3	P	5		CFOP6F	COD. FORNITORE/LAVORANTE
UNMI6F	58	59	2	A			UNMI6F	UNITA' DI MISURA 1=TINTO IN CAPO
FMLA6F	60	60	1	A			FMLA6F	3=COMMERCIAL.4=PUBBLICITA

CMP16F	61	63	3	A			CMP16F	1° COMPOSIZIONE
CMP26F	64	66	3	A			CMP26F	2° COMPOSIZIONE
CMP36F	67	69	3	A			CMP36F	3° COMPOSIZIONE
QQ056F	70	72	3	A			QQ056F	
CGFP6F	73	74	2	P	3		CGFP6F	GRUPPO FISCALE
PR016F	75	76	2	S	2		PR016F	ULTIMA RIGA MATERIALI
FL026F	77	77	1	S	1		FL026F	1=NO GESTIONE FISCALE
SCON6F	78	78	1	A			SCON6F	1=CREATI CONSUMI (DB AUTO)
FOMA6F	79	79	1	S	1		FOMA6F	FLAG OMAGGIO X ORD.E FATT.
MSG16F	80	80	1	A			MSG16F	FLAG MESSAGGIO
TA036F	81	82	2	S	2		TA036F	TAGLIA CAMPIONE
CSDL6F	83	84	2	A			CSDL6F	CODICE SIMBOLO DI LAVAGGIO
CDIS6F	85	88	4	A			CDIS6F	CODICE DISEGNO
PR026F	89	90	2	S	2		PR026F	RIGA DEL MATERIALE SOSTITUIBILE
.								
.								
.								
PTRA6F	255	7	3	S	3	1	PTRA6F	% CALO TRAMA STANDARD
PTRF6F	258	260	3	S	3	1	PTRF6F	% CALO TRAMA FORNITORE
PORD6F	261	263	3	S	3	1	PORD6F	% CALO ORDITO STANDARD
PORF6F	264	266	3	S	3	1	PORF6F	% CALO ORDITO FORNITORE
PEST6F	267	269	3	P	5		PEST6F	PESO DEL TAGLIATO
DIPR6F	270	277	8	S	8		DIPR6F	DATA INIZIO PRODUZIONE
LSAL6F	278	278	1	A			LSAL6F	S=LANCIATO A SALDO MODELLO
=====								
=====								

Lunghezza

Record: 278      Nr.Campi del file: 93

Fig. 4.7 Tracciato record di DBANA

## 4.2 Base di dati MySQL

La procedura Java realizzata va ad importare i dati estratti tramite l'ETL nella base di dati MySQL (vedi fig.4.1) . Questa base di dati è stata fornita da Quix s.r.l. (di seguito Quix). Si è scelto di utilizzare una copia di test, della base di dati attualmente in produzione, per motivi di sicurezza in quanto sull'originale operano applicazioni aziendali realizzate da Quix sulle quali lavorano sia utenti interni che esterni all'azienda. La base di dati utilizzata nel progetto si può dividere in 2 parti; la prima (vedi fig. 4.8) presenta le entità in cui andranno salvati i valori importati, la seconda (vedi fig. 4.9) rappresenta entità in cui si è definita la struttura dei tipi di dato e le proprietà di questi tipi.



## 4.2.1 Diagramma ER

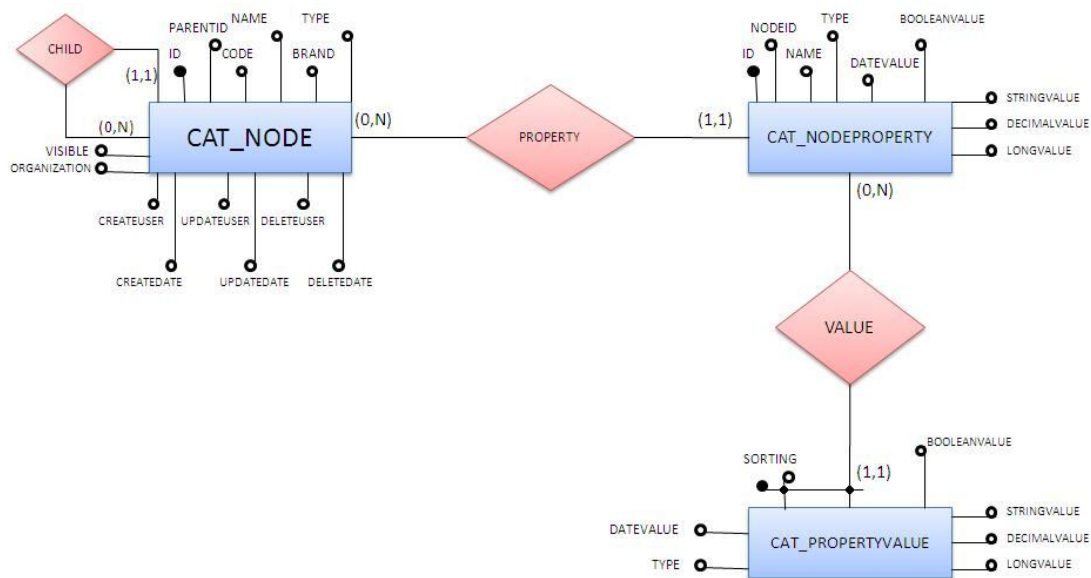


Fig. 4.8 Diagramma ER schema A

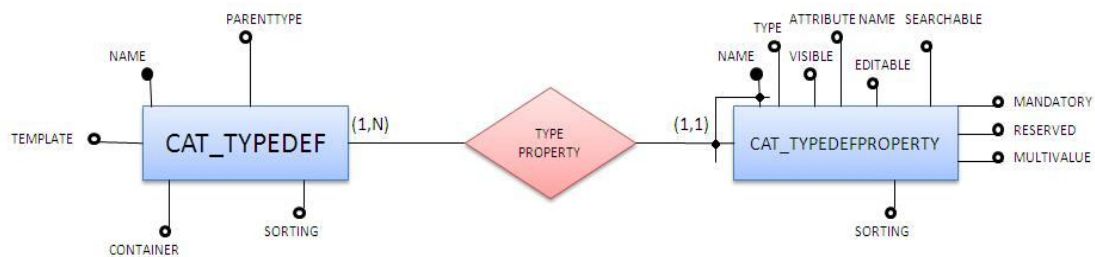


Fig. 4.9 Diagramma ER schema B

Di seguito viene riportata una descrizione accurata delle entità, delle relazioni e dei vincoli dei due schemi (A e B) presentati precedentemente che racchiude le regole che guidano la logica della base di dati.

Entità	Descrizione	Attributi	Identificatore
Cat_node	Contiene i tipi di entità definite nel dominio applicativo  (Stagione, Linea, Prodotto)	Id, Parentid, Code, Name, Brand, Type, Visible, Organization, Createuser, Createdate, Updateuser, Updatedate, Deleteuser, Deletedate	Id
Cat_nodeproperty	Contiene le proprietà dei tipi di entità (Es. per il prodotto il segnataglia, la composizione ecc)	Id, Nodeid, Name, Type, Datevalue, Booleanvalue, Stringvalue, Decimalvalue, Longvalue	Id
Cat_propertyvalue	Contiene le proprietà definite da liste come i colori o i listini prezzo	Sorting, Booleanvalue, Stringvalue, Decimalvalue, Longvalue, Datevalue, Type	Sorting, Id
Cat_typedef	Contiene la relazione tra le entità del dominio (Stagione padre di Linea che è padre di Prodotto)	Name, Parenttype, Template, Container, Sorting	Name
Cat_typedefproperty	Definisce tutte le proprietà dei tipi di entità	Name, Type, Visible, Attributename, Editable, Searchable, Mandatory, Reserved, Multivalued, Sorting	Name, Typedefname
Relazione	Descrizione	Entità Coinvolte	Attributi
Child	Associa un'entità padre all'entità figlia	Cat_node	
Property	Associa una entità alle sue proprietà	Cat_node, Cat_nodeproperty	
Value	Associa le proprietà alla lista di valori che la compongono	Cat_nodeproperty, Cat_propertyvalue	
Type Property	Associa i tipi alle sue proprietà	Cat_typedef, Cat_typedefproperty	

Riassumendo la tabella Cat\_node contiene le informazioni relativi ai nodi (Stagione, Linea, Prodotto) con la loro chiave, l'eventuale codice relativo al padre ed informazioni aggiuntive come data creazione, data update, data cancellazione. La tabella Cat\_nodeproperty contiene le proprietà relative ai nodi come segna taglia e prezzo per il nodo Prodotto, oppure venduto per il nodo Linea; mentre la tabella Cat\_propertyvalue contiene quelle proprietà relative ai nodi che sono contenute in una lista come i colori degli articoli, ai quali si accede con l'id della proprietà e la posizione nella lista (la chiave proprietà da sola non sarebbe bastata).

La struttura della base di dati MySQL presenta alcuni vantaggi rispetto a quella dell'AS400. Il primo vantaggio della struttura di destinazione riguarda la "portabilità" della base dello schema di arrivo, esso infatti può essere realizzato anche in altre basi di dati come ad esempio Postgres mantenendo la funzionalità delle API, altro vantaggio è che questo schema è indipendente dal contesto di utilizzo infatti la struttura potrebbe essere utilizzata sia in contesti come la metalmeccanica, l'edilizia o dove si desidera. Le interrogazioni avvengono su un numero limitato di tabelle rispetto alla base di dati Legacy di partenza.

La base di dati Legacy presenta ulteriori svantaggi rispetto a quella di arrivo:

a) Duplicazione di dati e pericoli di inconsistenza: come visto nel precedente paragrafo, nella base di dati iniziale può accadere che dati relativi ad un certo aspetto vengano utilizzati da applicazioni dell'ERP in modo "personalizzato" e che siano aggiornati in tempi diversi. A parte lo spreco di spazio, occorre stabilire complesse regole organizzative per far sì che ogni applicazione operi sui dati aggiornati.

b) Difficoltà delle modifiche alla definizione dei dati: se per qualche motivo, come spesso si rende necessario, si deve modificare la struttura dei record di uno o più file, si è costretti a riportare la modifica in tutte le applicazioni che utilizzano quei file. Ad esempio la modifica della lunghezza di un campo stringa o l'aggiunta di un nuovo campo ai record di un file, richiedono la modifica dei programmi che usano quei file. Senza contare la riconversione dei vecchi dati (storico) per tener conto delle modifiche. Nell'approccio della base di dati MySQL la proprietà di indipendenza logica assicura che non è necessario modificare i programmi che non fanno riferimento ai campi modificati o introdotti.

c) Dipendenza fisica: le operazioni per l'accesso ad un particolare file dipendono dall'organizzazione fisica che si è data all'archivio. Se si decide di cambiare organizzazione

occorre cambiare anche tutti i programmi che fanno riferimento a quell'archivio. L'indipendenza fisica è posta a fondamento della struttura in MySQL.

d) Necessità di soluzioni “ad hoc” per regolare accessi concorrenti: se per qualche motivo si vuole o si deve permettere l'accesso concorrente ad uno o più file dati occorre fare riferimento agli strumenti messi a disposizione dal linguaggio con cui si costruiscono le applicazioni trovando soluzioni personalizzate.

### 4.3 Estrazione dati

Tramite l'utilizzo di un ETL siamo andati ad esportare i dati dagli archivi dell'ERP legacy. Sono stati creati tre file .csv relativi rispettivamente alle stagioni, alle linee e agli articoli. I campi nei file .CSV sono separati dal carattere “;”. L'estrazione è stata fatta anche in file sequenziali ed in tabelle di basi di dati, ma dato che la procedura Java implementata è in grado di importare solamente dati di tipo .CSV, non vengono riportate in questo contesto.

Come ETL si è utilizzato Kettle che permette di eseguire Job formati da più trasformazioni. Il Job creato (vedi fig. 4.10) prevede tre trasformazioni in sequenza, la prima esporta le stagioni (season), la seconda le linee (line) e la terza gli articoli (articles). In caso di fallimento di una trasformazione viene mandata una email al responsabile IT con i log degli errori.

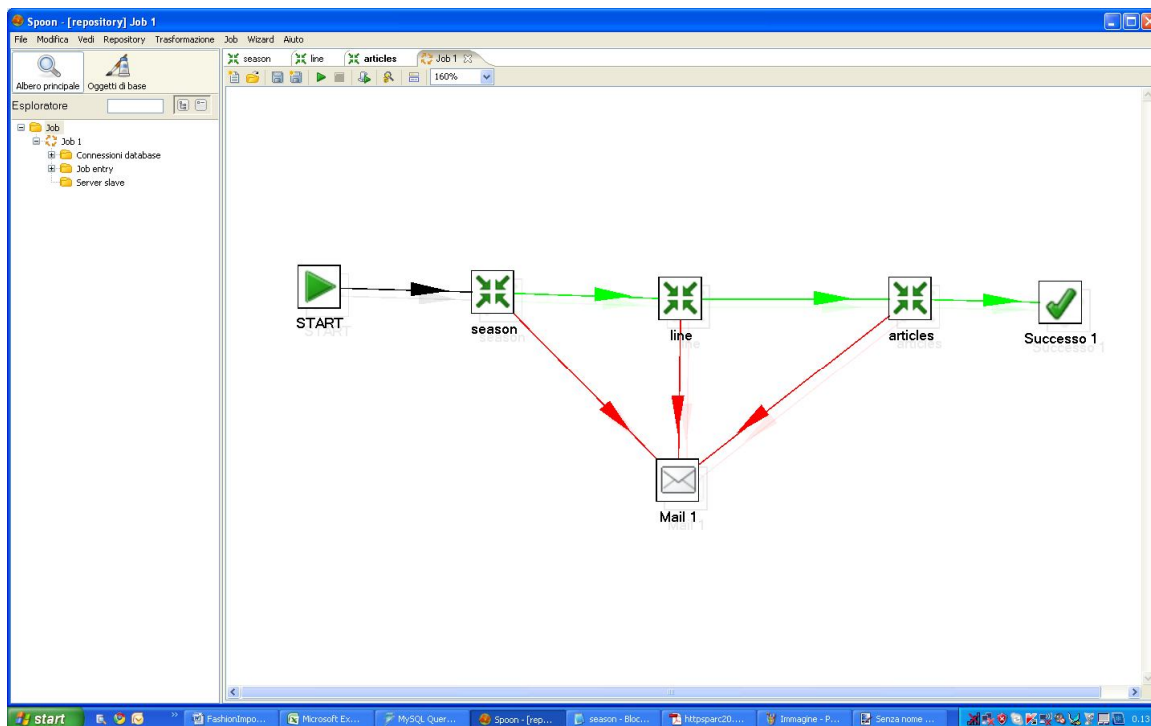


Fig. 4.10 Job ETL

Per le stagioni l'operazione è stata semplice in quanto si è interrogato un solo file, DSVSV, che contiene tutte le informazioni necessarie.

L'interrogazione SQL è la seguente:

```
select SREC03, substring(keyx03,4,9) AS KEYX03, SUBSTRING(DATI03,0,21) as dati03,  
substring(dati03,37,8)as datefrom, substring(dati03,50,8)  
as dateto from SICON_FIL.DSVAR where KEYX03 LIKE 'SV %' and SREC03 = 'A'
```

come si può notare la tabella dalla quale abbiamo estratto i dati è la DSVAR (Cfr. Par 4.1.3) che contiene la DSVSV ed il risultato è il seguente:

```
A;19;A/I 2006/2007 ;00000 ;050703  
A;20;P/E 2007 ;00000 ;060103  
A;21;A/I 2007/2008 ;00000 ;060707  
A;22;P/E 2008 ;00000 ;061215  
A;23;A/I 2008/2009 ;00000 ;070701  
A;24;P/E 2009 ;00000 ;080107  
A;25;A/I 2009/2010 ;00000 ;080702  
A;26;P/E 2010 ;00000 ;081201  
A;27;A/I 2010/2011 ;00000 ;090715  
A;28;P/E 2011 ;00000 ;100101  
A;8 ;P/E 2001 EX MC ;00000 ;030121  
A;99;NO ANALITICA ;00000 ;050103
```

I campi in successione rappresentano: stato record, codice stagione, descrizione, data inizio e data fine produzione.

Per le linee la trasformazione è stata simile ma si è interrogato i file DSVLI in DSVAR e DBANA (che contiene la relazione stagione linea) così da avere l'associazione corretta stagione-linea:

Es.

```
A;E1;E13;LES PARROTINES;S6;SEASON;273  
A;E1;E1M;BLUGIRL FOLIES;S6;SEASON;274  
A;E1;E15;MISS BLUMARINE;S6;SEASON;274  
A;28;282;PARROT;S6;SEASON;274  
A;28;285;MISS BLUMARINE;S6;SEASON;275  
A;E1;E16;ICEBERG;S6;SEASON;275  
A;E1;E17;BABY BLUMARINE;S6;SEASON;275  
A;E1;E1K;KI6;S6;SEASON;275  
A;E1;E1I;ICEBERG BABY;S6;SEASON;276  
A;28;28K;KI6;S6;SEASON;276  
A;28;283;LES PARROTINES;S6;SEASON;276  
A;28;28M;BLUGIRL FOLIES;S6;SEASON;277  
A;E1;E1N;NEONATO BLUMARINE;S6;SEASON;277
```

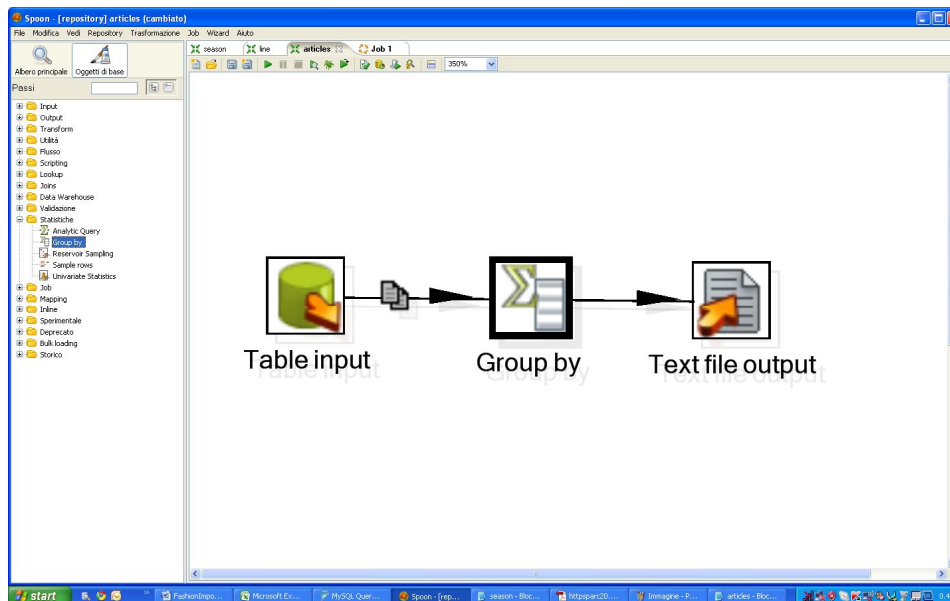
A;28;287;BABY BLUMARINE;S6;SEASON;277  
A;E1;E19;LU-MA;S6;SEASON;277  
A;27;27A;BLUMARINE SCARPE;S6;SEASON;278

Qui i campi sono: stato record, stagione riferimento, codice linea, descrizione, organizzazione, padre e venduto. Da sottolineare sono i campi “stagione riferimento” e “codice linea”. Il primo servirà all’applicazione Java per capire chi è la stagione padre della linea a cui fa riferimento (e settarne l’id), il secondo è ottenuto dalla concatenazione di codice stagione e codice linea contenuti in DBANA che facilitano così il reperimento di una linea in una stagione.

Per gli articoli si sono interrogati i file DBANA in Join con DSCOL (che contiene colori relativi agli articoli):

```
SELECT DISTINCT SREC6F, concat(CSTA6F,CLIN6F),  
concat(CSTA6f,concat(clin6f,concat(CMOD6F,CART6F))) as code, des16f, cset6f,  
ccac6f, cmp16f, cmp26f, cmp36f, 'S6' as ORG,  
'LINE' as LINE, cvar0g as colore , PUNI6F from DRESS_COM.DBANA f join  
DRESS_COM.DSCOL  
ON f.Csta6F = CSTA0G AND F.CLIN6F = CLIN0G AND F.CMOD6F = CMOD0G AND F.CART6F =  
CART0G WHERE F.srec6f = 'A' and cart6f != 'PUBB'  
order by code , colore
```

Si può notare che data la struttura della base di dati dell’ERP anche nel file dei colori DSCOL si accede riprendendo la chiave dell’articolo formata da codice stagione, linea, articolo più il codice variante colore. La difficoltà in questo caso è data dal fatto che questa query individua una riga per ogni combinazione “stagione-linea-articolo-colore” mentre si voleva avere tutti i codice colore di un articolo in una unica lista all’interno di un campo così da semplificare l’importazione della procedura in JAVA e fare meno accessi alla base di dati. Per fare questo si è aggiunta una funzione di Kettle chiamata “Group By” (Vedi fig. 4.11).



Restituisci sempre una riga di risultato

Campi che formano il gruppo:

?	▲	Campo di raggruppamento	
1		SREC6F	
2		00002	
3		CODE	
4		DES16F	▼
5		CSET6F	
6		CCAC6F	
7		CMP16F	
8		CMP26F	
9		CMP36F	
10		ORG	
11		LINE	
12		PUNI6F	

Aggregati:

?	▲	Nome	Oggetto	Tipo	Valore
1		COLORE	COLORE	Concatena le string separate da	
2					

Fig. 4.11 Trasformazione con Group By

Come riporta la figura 4.11, sotto la voce “Campi che formano il gruppo” si vanno ad aggiungere tutti i campi prelevati con la query in “table input” tranne quelli che si vogliono raggruppare in un unico campo, che vanno aggiunti sotto la voce “aggregati” inserendo alla voce “nome” il nome del nuovo campo formato dalla lista, sotto “Oggetto” va inserito il campo da aggregare, selezionando il tipo di operazione e un eventuale valore utilizzato come separatore.

Il risultato è il seguente:

A;26M;26M8001 ;GIUBBINO LINO SPALMA;I;17 ; RZ; ; ;S6;LINE;10|527;644  
A;26M;26M8003 ;SPOLVERINO C/FRAPPE ;I;15 ; J; U3; ;S6;LINE;99|508|531|547;716  
A;26M;26M8004 ;GIUBBINO TECNICO ;I;8 ; J; GZ; ;S6;LINE;10|99;498  
A;26M;26M8005 ;GIACCA PELLE SFIANC.;I;31 ; BF; DC; ;S6;LINE;99|538|542|546;730  
A;26M;26M8007 ;GIUBBINO A BUSTIER ;I;33 ; D; ; ;S6;LINE;10|99|520|527|539|542;524  
A;26M;26M8007 1;GIUBBINO C/CINTURA ;I;33 ; D; ; ;S6;LINE;10|99|520|527|539|542;618  
A;26M;26M8010 ;GIUBBINO JEANS ;I;V.U.; D; ; ;S6;LINE;517|527|888;540  
A;26M;26M8011 ;GILET RICAMATO ;I;16 ; C; ; ;S6;LINE;10|99|546;200  
A;26M;26M8012 ;SPOLVERINO C/FRAPPE ;I;V.U.;A8P;A5H; ;S6;LINE;546;850

I campi prelevati rappresentano: Stato record, codice padre dell'articolo (secondo l'albero visto nel paragrafo 4.2), una superchiave formata dalla concatenazione di codice stagione, linea ed articolo che va a formare un unico codice articolo, la descrizione, il codice segna taglia, il codice cartella colori, 3 campi di composizione, l'organizzazione, la tipologia del padre, la lista dei codice colore (evidenziata) e il venduto per articolo.

#### 4.4 Definizione della tabella di mapping

La tabella di mapping consiste in una tabella chiamata Cat\_proceduredescriptor contenuta nella base di dati di destinazione. Questa tabella è necessaria per mappare i dati di origine fornendo così alla procedura di importazione una logica per importare i dati.

Ogni campo è impostato seguendo le regole principali di esportazione ricavate dalle tre tipologie di sorgente dati: file sequenziale, file a separatore e tabella di dati. Per creare questa tabella quindi si sono analizzate le proprietà di file sequenziali, file a separatore e tabelle di dati. Di seguito l'elenco dei campi della tabella con una descrizione di tale scelta ed impiego nel progetto:



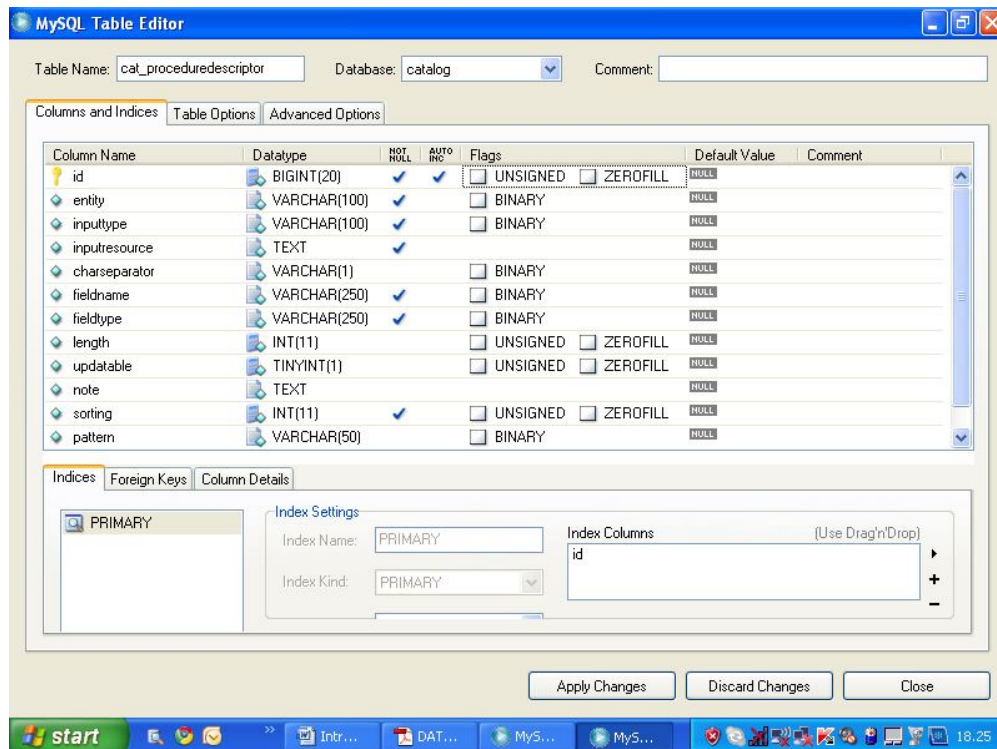


Fig. 4.12 Struttura tabella Cat\_proceduredescriptor

Id: è la chiave primaria di ogni campo e viene generata automaticamente all'aggiunta di un record nella tabella.

entity: questo campo serve alla procedura di importazione come indicazione su qual è l'entità di riferimento se Stagione, Linea o Articolo. Tramite questo campo la procedura associa le informazioni successive all'entità menzionata.

inputtype: indica la sorgente dati che può essere SEPARATED\_FILE per il file a separatore, FIXED\_FILE per il file a lunghezza prefissata e TABLE per la tabella di database.

inputresource: indica dove risiedono le informazioni da importare, nel caso dei file a separatore ed a lunghezza fissa contiene il percorso del file.

charseparator: questo campo viene compilato solo nel caso di file a separatore ed indica alla procedura quale sarà il carattere separatore dei vari campi in ogni record nel file.

fieldname: indica il nome del campo che si andrà ad importare, viene utilizzato dalla procedura per capire di che campo si tratta (es. codice, descrizione, prezzo ecc) ed effettuare le relative operazioni di importazione nella base di dati. Ad esempio il campo "stato\_record"

indica che l'informazione è relativa allo stato record, nel caso il valore fosse "C" di cancellato la procedura andrà a settare i campi "deleteuser" e "deletedate" in quanto si tratta di cancellazione logica e non di cancellazione fisica dei record.

fieldtype: indica il "Tipo di dato" relativo al campo, attraverso questo campo la procedura setterà il tipo nella base di dati di destinazione ed eventualmente prima di questa operazione effettuerà delle operazioni di formattazione su tale campo.

length: indica la lunghezza del campo, viene utilizzato dalla procedura nel caso la sorgente sia un file a lunghezza prefissata per permettere alla procedura di importazione di conoscerne la lunghezza.

updatable: questo campo indica se l'informazione dovrà essere aggiornata o meno nella base di dati di destinazione, attualmente non viene considerato nella procedura ma viene ritenuto utile per uno sviluppo futuro.

note: serve per aggiungere eventuali note riguardanti il campo da importare, anche in questo caso la procedura non considera questo campo.

sorting: indica la posizione consecutiva del campo rispetto agli altri attributi relativi all'entità e viene utilizzato dalla procedura nel caso di importazione da FIXED FILE.

pattern: questo campo allo stato dell'arte non viene utilizzato dalla procedura di importazione ma risulterà utile negli sviluppi futuri per indicare la formattazione del campo a cui fa riferimento, ad esempio se il campo è di tipo DATE all'interno di pattern si potrà precisare alla procedura di importazione la formattazione della data "yyyymmdd" oppure "ddMMyyyy".

Esempio di tabella compilata con sorgenti per Stagioni, Linee e Prodotti file a separatore .CSV:

id	entity	inputtype	inputresource	charseparator	fieldname	fieldtype	length	updateable	note	sorting	pattern
25	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	stato_record	STRING	1	1	NULL	0	NULL
26	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	line	STRING	2	1	NULL	1	NULL
27	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	code	STRING	3	1	NULL	2	NULL
28	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	name	STRING	0	1	NULL	3	NULL
29	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	size_group	STRING	0	1	NULL	4	NULL
30	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	category	STRING	0	1	NULL	5	NULL
31	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	composition_1	STRING	0	1	NULL	6	NULL
32	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	composition_2	STRING	0	1	NULL	7	NULL
33	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	composition_3	STRING	0	1	NULL	8	NULL
34	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	organization	STRING	0	1	NULL	9	NULL
35	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	parent_type	STRING	0	1	NULL	10	NULL
36	PRODUCT	SEPARATED_FILE	C:\TESIVARTICLE.CSV	;	color	COLLECTION	0	1	NULL	11	NULL
37	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	stato_record	STRING	0	1	NULL	0	NULL
38	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	season	STRING	0	1	NULL	1	NULL
39	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	code	STRING	0	1	NULL	2	NULL
40	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	name	STRING	0	1	NULL	3	NULL
41	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	organization	STRING	0	1	NULL	4	NULL
42	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	parent_type	STRING	0	1	NULL	5	NULL
43	LINE	SEPARATED_FILE	C:\TESIVLINE.CSV	;	quantity	LONG	0	1	NULL	6	NULL
44	SEASON	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	stato_record	STRING	1	1	NULL	0	NULL
45	SEASON	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	code	STRING	2	1	NULL	1	NULL
46	SEASON	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	name	STRING	20	1	NULL	2	NULL
47	SEASON	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	date_from	DATE	8	1	NULL	3	NULL
48	SEASON	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	date_to	DATE	8	1	NULL	4	NULL
49	PRODUCT	SEPARATED_FILE	C:\TESIVSEASON.CSV	;	cost_1	DECIMAL	0	1	NULL	12	NULL

Fig. 4.13 esempio tabella Cat\_proceduredescriptor compilata

Questa tabella ha alcuni vincoli importanti che riguardano la compilazione del campo fieldname:

1. il primo campo di ogni entità deve essere “stato\_record”;
2. il secondo indica il codice del padre e deve essere per la linea “season” e per il prodotto “line”. La stagione non ha padre, quindi non è soggetta a questo vincolo.
3. Il terzo campo per la linea e il prodotto è il campo “code” che indica la chiave;
4. Il quarto è il campo “name” indica la descrizione.

I vincoli di posizione riguardano espressamente linea e prodotto (a parte il primo “stato\_record), mentre per tutte le entità è importante mantenere la formattazione evidenziata: “stato\_record”, “season”, “line”, “code” e “name”, il resto dei nomi non ha vincoli. Devono essere impostati in questo modo perché la procedura Java ha al suo interno logiche specifiche di importazione.

## 4.5 Implementazione della procedura Java di importazione dati

La procedura java realizzata sviluppa solo l'importazione da file .CSV, l'importazione da altre sorgenti (file sequenziale e tabella) è rimandata a sviluppi futuri. Le API di collegamento alla base di dati MySQL sono state fornite da Quix.

La procedura è organizzata in quattro package:

1. `package it.quix.fashion.importer.test;`
2. `package it.quix.fashion.importer.objectpopulator;`
3. `package it.quix.fashion.importer.utils;`
4. `package it.quix.fashion.importer.constant;`

Nel package `it.quix.fashion.importer.test` è contenuta la classe `ProcedureDescriptorTest` che ha al suo interno il main. Scopo di questa classe è reperire dalla tabella di mapping le informazioni riguardanti i nodi da importare (Stagioni, Linee ed Articoli), queste informazioni vengono salvate in una lista che viene passata ad un'altra classe che andrà a reperire le informazioni dai file .CSV restituendo i nodi compilati che verranno salvati nella base di dati mySQL.

Nel package `it.quix.fashion.importer.objectpopulator` vi è la classe `ObjectPopulator`, che riceve la lista dei nodi dalla classe `ProcedureDescriptorTest`. Questa classe tramite le informazioni ricevute, tra cui il percorso dei file e la lista dei campi per ogni file (es. stato record, codice stagione/linea/prodotto), va a recuperare le informazioni nei file e le inserisce record per record all'interno di un nodo, ogni nodo contiene le informazioni di un record. La lista dei nodi compilati vengono restituiti alla classe `ProcedureDescriptorTest` che li andrà a salvare.

Il package `it.quix.fashion.importer.utils` contiene la classe `importerUtils` che opera operazioni di trasformazione delle informazioni reperite nei record dei file. Ad esempio converte il formato data contenuto nei record, `yyyymmdd` in `ddmmyyyy`, adatto alla base di dati MySQL.

Il package `it.quix.fashion.importer.constant` contiene la classe `Costants` che definisce alcune costanti utilizzate dalle altre classi.

## 4.6 Strumenti utilizzati

In questo capitolo vengono brevemente illustrati gli strumenti adottati, le scelte tecnologiche adottate e l'ambiente di sviluppo prescelto per l'analisi e l'implementazione del progetto.

### 4.6.1 DBVisualizer

La prima fase del progetto è stata lo studio del data base DB2 sul quale lavora l'ERP aziendale, per analizzare le tabelle di interesse ho utilizzato il programma DBVisualizer. DBVisualizer consente di effettuare connessioni simultanee a differenti database attraverso i drivers JDBC. Dispone di un'interfaccia grafica intuitiva che permette di visualizzare rapidamente la struttura dei database, di ottenere caratteristiche dettagliate sugli oggetti, editare tabelle grafiche con i dati raccolti, eseguire script o statement SQL. Il prodotto è distribuito in due versioni: DBVisualizer free, gratuito, sia per uso non commerciale che commerciale con le funzioni base e DBVisualizer personal, per uso commerciale con licenza, che potenzia la versione free con la possibilità di editare tabelle di dati, eseguire script SQL, gestire i dati BLOB/CLOB e fornisce supporto per la creazione di tabelle e indici.

Caratteristiche:

- Supporto Oracle, DB2, Mimer, SQL Server, Sybase ASE, Informix, MySQL, PostgreSQL e JavaDB / Derby
- Creazione guidata Database di connessione
- Connessioni al database organizzate in cartelle
- Supporta connessioni multiple simultanee ai database
- Struttura ad albero per cercare oggetti di database
- Visualizzare le informazioni delle tabelle, indici, chiavi primarie, i privilegi, gli schemi, le banche dati, procedure e molto di più
- Ricerca di oggetti nei database
- Esportazione di dati in SQL o file XML
- Visualizzare formati BMP, TIFF, PNG, GIF e JPEG
- Tipo di rilevazione automatica dei dati
- Impostare proprietà univoche per ogni connessione al database

Per scaricare la versione free e per maggiori approfondimenti segnalo il sito [10].

## 4.6.2 Kettle

Una volta analizzato le tabelle del DB2 ho fatto l'esportazione dei record scelti nei formati richiesti dal progetto, per far questo ho utilizzato un motore ETL. Un ETL è un applicativo in grado di estrarre i dati dal database a cui si appoggia l'ERP, trasformarli e ripulirli se necessario ed infine esportarli in un file o in una base di dati; nel caso specifico l'esportazione dei dati è stata fatta in file .csv e .txt. Ho scelto un ETL open source Pentaho Data Integration versione 3.2.0 comunemente chiamato KETTLE o Spoon.

Pentaho Data Integration (PDI) dispone di potenti capacità di **estrazione, trasformazione e caricamento** (ETL - Extraction, Transformation and Loading), è composto da un insieme di strumenti che permettono di trasferire e manipolare i dati tra varie fonti, tipicamente DBMS e files nei vari formati. PDI utilizza un approccio innovativo basato sulla definizione di metadati per la descrizione ed il salvataggio dei processi di ETL, è indipendente dalla piattaforma ed ha un'architettura basata su standard internazionali. Dispone inoltre di un'interfaccia grafica di progetto molto intuitiva, di tipo "drag-and-drop", a vantaggio della produttività.

PDI è uno strumento di 4° generazione, frutto di una ormai lunga evoluzione del software design:

- 1° generazione: programmazione manuale, grosse quantità di codice prodotte, difficile da mantenere e da rilasciare.
- 2° generazione: codice generato una sola volta, necessità di modifica manuale, facile da generare, difficile da mantenere e rilasciare.
- 3° generazione: generazione del codice a partire da un modello, facilità di generazione, facilità di manutenzione, ma ancora difficile da rilasciare
- 4° generazione: possibilità di eseguire attività (task) direttamente dal modello, facilità di generazione, di rilascio e di manutenzione.

Per maggiori informazioni visitare il sito [11] mentre [12] fornisce un ottimo approfondimento sull'utilizzo di Kettle.

### 4.6.3 MySQL Administrator e MySQL Query Browser

La compilazione del tracciato standard è stata fatta manualmente. Per fare questo ho fatto uso dei principali strumenti che MySQL offre ad un progettista:

Fonte [8]:

4. MySQL Administrator è un programma per effettuare operazioni di amministrazione, come configurare il un server MySQL, tenere d'occhio il suo stato e le performance, avviarlo e fermarlo, effettuare backup, ed una serie di altri compiti di gestione. MySQL Administrator offre dei vantaggi tra cui:
  - La sua interfaccia grafica lo rende più intuitivo da usare.
  - Fornisce una migliore visione d'insieme sulle impostazioni che sono di vitale importanza per le performance, l'affidabilità e la sicurezza del server MySQL.
  - Mostra degli indicatori grafici di performance, rendendo così più facile determinare e regolare le impostazioni del server
  
5. MySQL Query Browser è uno strumento grafico fornito da MySQL AB per creare, eseguire ed ottimizzare query in ambiente grafico. MySQL Query Browser è progettato per aiutarvi ad effettuare query ed analizzare i dati memorizzati nel vostro database MySQL. Se da una parte tutte le query eseguite in MySQL Query Browser possono essere effettuate anche tramite l'utility da riga di comando `mysql`, MySQL Query Browser consente di eseguire query e modificare dati in maniera più intuitiva, tramite un'interfaccia grafica. MySQL Query Browser è progettato per funzionare con MySQL versione 4.0 e superiori. MySQL Query Browser è prevalentemente il risultato del feedback che MySQL AB ha ricevuto su un periodo di molti anni da molti utenti. Esiste un forum dedicato a MySQL Query Browser, che si trova al link <http://forums.mysql.com/list.php?108>.

#### 4.6.4 Il framework Eclipse

Il Framework di supporto adottato per la creazione della procedura di importazione dati è Eclipse [13] ;

Di seguito una descrizione del Framework tratta da [14]:

**Eclipse** è un progetto open source legato alla creazione e allo sviluppo di una piattaforma di sviluppo ideata da un consorzio di grandi società quali Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP e Serena Software, chiamato Eclipse Foundation, e creata da una comunità strutturata sullo stile dell'open source. Pur essendo orientata allo sviluppo del progetto stesso, questo IDE (ambiente di sviluppo integrato) è utilizzato anche per la produzione di software di vario genere. Si passa infatti da un completo IDE per il linguaggio Java (JDT, "Java Development Tools") ad un ambiente di sviluppo per il linguaggio C++ (CDT, "C/C++ Development Tools") e a plug-in che permettono di gestire XML, PHP e persino di progettare graficamente una GUI per un'applicazione JAVA (Eclipse VE, "Visual Editor"), rendendo di fatto Eclipse un ambiente RAD. Il programma è scritto in linguaggio Java, ma anziché basare la sua GUI su Swing, il toolkit grafico di Sun Microsystems, si appoggia a SWT, librerie di nuova concezione che conferiscono ad Eclipse un'elevata reattività. La piattaforma di sviluppo è incentrata sull'uso di plug-in, delle componenti software ideate per uno specifico scopo, per esempio la generazione di diagrammi UML, ed in effetti tutta la piattaforma è un insieme di plug-in, versione base compresa, e chiunque può sviluppare e modificare i vari plug-in. Nella versione base è possibile programmare in Java, usufruendo di comode funzioni di aiuto quali: completamento automatico ("Code completion"), suggerimento dei tipi di parametri dei metodi, possibilità di accesso diretto a CVS e riscrittura automatica del codice (funzionalità questa detta di Refactoring) in caso di cambiamenti nelle classi. Essendo scritto in Java, Eclipse è disponibile per le piattaforme Linux, HP-UX, AIX, Mac OS X e Windows.



## 4.6.5 Linguaggio di programmazione

In riferimento al linguaggio la scelta è ricaduta su Java. Le caratteristiche principali di questo linguaggio sono:

- Indipendenza dalla piattaforma cioè capacità di “muoversi” da un calcolatore ad un altro in modo semplice.
- Nel codice sorgente
  - dati primitivi sono definiti con dimensioni consistenti su tutte le piattaforme
  - librerie (classi) per le funzioni di base
- Nel codice binario
  - Bytecodes, codifica “intermedia” tra codice sorgente e codice eseguibile, interpretata sulla macchina client
- Programma orientato agli oggetti
- Organizzazione del software
- Programmazione modulare
- Riutilizzo del codice
- Semplicità
- Rinuncia ai puntatori
- Stringhe ed array sono oggetti
- Gestione della memoria automatica

Per maggiori informazioni è possibile consultare il sito ufficiale [15], oppure [16] e [17].

## 5. CONCLUSIONI E SVILUPPI FUTURI

Il primo obiettivo del progetto era l'estrazione dei dati dalla base dati dell'ERP Legacy ed il salvataggio rispettivamente in file sequenziali, file a separatore e tabella di base di dati, che avrebbero formato l'origine dei dati per la procedura Java. Altro scopo dell'estrazione era quello di trasformare i dati di origine per semplificare le logiche di importazione nella base di dati MySQL. Ad esempio per gli articoli, le informazioni dei colori risiedevano in più file differenti nell'ERP, tramite la procedura di estrazione sono stati consolidati in un unico campo "colore" più semplice da gestire. La fase di estrazione dati è dipendente dall'ERP di origine, se l'azienda decidesse in futuro di sostituirlo, per la procedura di importazione non cambierebbe nulla, in quanto utilizza sorgenti "preparate" dall'ETL che sarà l'unico processo che andrà modificato. La modifica di un processo di ETL risulta più semplice (nel caso debba solo estrarre i dati) del riscrivere il codice Java nella procedura. Un sotto obiettivo dell'estrazione era lo studio di un programma che permettesse di estrarre i dati; si è scelto l'ETL Kettle per la sua semplicità di utilizzo. Il primo obiettivo è stato raggiunto con successo.

Una volta completata la fase di estrazione il secondo obiettivo riguardava la creazione di una tabella di mapping per l'esportazione dei dati. La realizzazione della tabella è stata possibile attraverso le seguenti fasi:

1. Analisi delle caratteristiche specifiche di quelle che sono state individuate come sorgenti dati per l'importazione (file a lunghezza prefissata, file a separatore e database), per individuare le logiche di esportazione dati da ogni sorgente;
2. Analisi delle caratteristiche specifiche della base di dati di destinazione (MySQL);
3. Implementazione della tabella di mapping strutturata in base ai requisiti emersi nel punto 1 e 2.

Questa tabella rappresenta la parte più importante del progetto, insieme alla procedura Java, perché permette di importare facilmente nuove informazioni. L'inserimento di un nuovo campo, avviene semplicemente aggiungendo alla tabella di mapping un record che ne definisca le proprietà, senza dover modificare la procedura Java.

Per quanto concerne il terzo obiettivo, la procedura Java è in grado di importare dati da sorgente di tipo File a separatore .CSV. L'aggiunta di campi, da importare in diverse fasi, è andata a buon fine. La difficoltà maggiore è stata quella di comprendere ed utilizzare le API fornite dalla Quix che andavano ad agire sulla base di dati di destinazione anch'essa fornita da Quix. L'obiettivo si considera raggiunto.

La realizzazione di altre funzioni nella procedura Java per l'importazione dalle altre due sorgenti dati (file sequenziale e tabella) definite negli obiettivi fornisce uno spunto importante per eventuali sviluppi futuri. Ulteriori sviluppi si potranno implementare sfruttando i campi della tabella di mapping che attualmente non vengono utilizzati dalla procedura Java; da sottolineare il campo "note" ed il campo "pattern" utile ad esempio per indicare la formattazione dei dati (Cfr. Par. 4.4), per esempio la procedura attuale implementa solamente il tipo di data "yyyymmdd" fornito dall'ERP di SpazioSei trasformandolo in "ddmmyyyy" e la logica della trasformazione è contenuta nel codice; sfruttando il campo "pattern" si potrà rendere ancora più flessibile e standard l'importazione. Un ulteriore sviluppo riguarda il campo "updatable", come descritto nel paragrafo (Cfr. Par. 4.4), sfruttando questo campo si potrà evitare che, dati residenti nella basi di dati MySQL, che non dovranno essere sovrascritti dall'esportazione, vengano modificati.

La compilazione della tabella di mapping attualmente avviene utilizzando MySQL Query Browser o programmi specifici, questo risulta non "user friendly" si potrebbe quindi realizzare un' interfaccia grafica che ne permettesse la compilazione e la modifica in modo semplice per facilitare utenti meno esperti.

L'IT interno di SpazioSei ha tratto svariati benefici dalla realizzazione di questo progetto:

- lo studio delle strutture dati di ERP e MySQL, sulla quale lavorano le applicazioni che SpazioSei ha interfacciato, ha portato a comprendere la complessità che risiede in ogni sviluppo, favorendo così la capacità, di valutazione di preventivi ricevuti dai propri fornitori.
- La comprensione della complessità degli sviluppi consentirà di "filtrare" ed inquadrare le richieste degli utenti aziendali (in termini di attività) per fornire così, informazioni specifiche e mirate ai fornitori, evitando inutili incomprensioni, perdite di tempo e denaro.

- Il personale IT interno a SpazioSei, ha acquisito maggiore autonomia, ora ogni fornitore si dovrà occupare solamente del proprio applicativo, lasciando all'IT aziendale il compito di estrazione dei dati dal gestionale e la loro importazione nella base di dati MySQL. Questo in precedenza non era possibile in quanto, essendo l'estrazione basata su procedure di ETL realizzate "ad hoc", ogni aggiunta di record da importare implicava l'intervento di programmatori o, nei casi più complessi, di analisti, da parte dei fornitori con conseguenti costi per l'azienda.

Dal punto di vista amministrativo l'utilizzo di una procedura di importazione standard quindi potrà consentire un risparmio economico per l'azienda; a cui va aggiunto l'utilizzo di MySQL che ha già abbattuto i costi di gestione.

Il progetto ha dimostrato che operazioni di interfacciamento tra applicazioni, troppo spesso ritenute "banali" o scontate dal Management, danno la possibilità alle aziende di sviluppare il proprio mercato a livello mondiale. I grandi progetti di Business dai quali traggono beneficio centinaia di persone (come nel caso di SpazioSei) hanno alla base "piccole" attività come quelle descritte in questa tesi.

## 6. BIBLIOGRAFIA

- [1] Dayani – Fard, H e altri, “Legacy Software Systems: Issues, Progress, and Challenges” IBM Technical Report: TR-74.165-k, Aprile 1999,  
<https://www-927.ibm.com/ibm/cas/publications/TR-74.165/k/legacy.pdf>
- [2] Antonio Massari-Massimo Mecella, “*IL TRATTAMENTO DEI LEGACY SYSTEM*”, Dispense, <http://corsi.deis.unical.it/OLD/sistemiInformativi/upload/dispense/42.pdf>, Anno 2002
- [3] Cipriano Egidio – Teti Antonio “Eucip – Il manuale per l’informatico professionista. Certificazione Core Level” Hoepli, Anno 2005.
- [4] “IBM System i5, eServer i5, and iSeries Systems Builder IBM i5/OS Version 5 Release 4” , Anno 2006, <http://www.redbooks.ibm.com/abstracts/sg242155.html?Open>
- [5] Guerzoni Fabio, Cucchiara, Rita ”Sistemi gestionali: da As400 a BaaN” , Università degli Studi di Modena e Reggio Emilia, sede di Modena, Facoltà di Ingegneria, Corso di laurea in Ingegneria informatica, Anno Accademico 2002-2003.
- [6] Gianluigi Di Donato, “DBMS – 3: MySQL”, articolo pubblicato sul sito “Ingegneri pesanti”, 7 Febbraio 2007, <http://ingegneripesanti.wordpress.com/category/database-e-dbms/>.
- [7] DB2 Workload Manager for Linux, UNIX, and Windows, <http://www.redbooks.ibm.com/abstracts/sg247524.html?Open>
- [8] sito ufficiale MySQL, <http://www.mysql.org/>
- [9] Entela Kazazi , Sonia Bergamaschi. “Confronto tra microsoft SQL server 2000 e MySQL 5.0” Anno 2006, Università degli studi di Modena e Reggio, facoltà di Ingegneria, sede di Modena, Corso di laurea in Ingegneria Informatica - Nuovo Ordinamento. - Anno Accademico 2005/2006.
- [10] Sito DBVisualizer, <http://www.dbvis.com/products/dbvis/download/>
- [11] Sito Kettle, <http://kettle.pentaho.org>

[12] Dannaoui Abdul Rahman, Bergamaschi Sonia, “\*Progetto e realizzazione di tecniche di business intelligence per l'analisi ambientale” , Università degli studi di Modena e Reggio, facoltà di Ingegneria, sede di Modena, corso di laurea specialistica in ingegneria informatica, A.A. 2008/2009

[13] Sito ufficiale Eclipse, <http://www.eclipse.org/>

[14] Wikipedia Eclipse, [http://it.wikipedia.org/wiki/Eclipse\\_%28informatica%29](http://it.wikipedia.org/wiki/Eclipse_%28informatica%29)

[15] sito ufficiale java: <http://java.sun.com/>

[16] Claudio De Sio Cesari “ Java. Mattone dopo mattone” Hoepli, Anno 2002, <http://www.claudiodesio.com/>

[17] Massimiliano Tarquini- Alessandro Ligi “ Manuale di Java 6. Programmazione orientata agli oggetti con Java”, Hoepli, Anno 2006

[18] [www.spaziosei.it](http://www.spaziosei.it)

## RINGRAZIAMENTI

*Il primo pensiero va ai miei genitori e alle mie sorelle per il sostegno che mi hanno sempre dato perché senza di esso non avrei raggiunto questo importante traguardo. Ringrazio Ilaria perché con amore e pazienza mi ha permesso di trovare forza ed energia mentale per affrontare al meglio l'ultima parte dei miei studi. Grazie anche a tutti i miei amici con i quali ho potuto sdrammatizzare le situazioni più difficili e che ho sempre sentito vicini e disponibili ad aiutarmi in ogni momento. Un ricordo speciale va a Filippo Saetti che porto sempre nel cuore.*

*Ringrazio la Professoressa Sonia Bergamaschi per l'aiuto durante lo svolgimento della tesi. Un ringraziamento particolare va all'Ing. Roberto Pellegrino, per il suo prezioso aiuto morale e scientifico e per la sua disponibilità e pazienza nei miei confronti.*

*Ringrazio l'azienda SpazioSei s.r.l. per la quale lavoro, perché mi ha dato la possibilità di fare questo progetto di tesi e mi sta dando l'opportunità di crescere mostrando fiducia nei miei confronti. Grazie anche a Quix s.r.l. per avermi messo a disposizione strumenti e persone veramente preparate con le quali è stato un piacere ed un onore collaborare.*

*Un grazie a tutte le band in cui suono (e con cui ho suonato), con le quali posso provare a realizzare i miei sogni, non posso, quindi, non ringraziare la musica perché per me è vera fonte di felicità e di vita.*

*Grazie veramente a tutte le persone che ho incontrato nella mia strada e con le quali ho potuto camminare, ognuna di esse mi ha insegnato qualcosa di importante che non potrò dimenticare...*

*Luca Torreggiani*