



## **UNIVERSITÀ DI PISA**

Facoltà di Economia  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea Magistrale in Informatica per l'economia e per l'azienda  
(Business Informatics)

### **TESI DI LAUREA**

PROGETTAZIONE E REALIZZAZIONE DI UN SISTEMA DI BI  
PER UN'AZIENDA DELLA DISTRIBUZIONE ALIMENTARE

### **RELATORI**

PROF. SALVATORE RUGGERI  
DOTT. STEFANO CELATI

### **CANDIDATO**

PASQUALE ROMEO

ANNO ACCADEMICO 2011-2012

*Alla mia famiglia  
per non avermi tagliato i viveri...*

## SOMMARIO

Nel lavoro di tesi vengono affrontate le tematiche inerenti le fasi di realizzazione di un sistema di Business Intelligence per le attività di vendita di un'azienda che commercia nel settore alimentare. L'obiettivo è di realizzare un contenitore unico per l'integrazione dei dati operazionali che fornisca un adeguato supporto all'impiego di una Business Intelligence riconciliata e coerente a tutte le divisioni aziendali. Vengono presentate le diverse fasi affrontate durante la realizzazione del *data warehouse*: lo studio dei processi, l'analisi e la specifica dei requisiti, la progettazione concettuale e logica dei *data mart* interessanti, l'analisi degli strumenti software utilizzati, la realizzazione delle procedure di estrazione, trasformazione e caricamento, la creazione dei cubi OLAP e la realizzazione di *query* MDX. Di ciascuna fase si presentano sia le problematiche di ordine generale, sia le soluzioni ai problemi riscontrati durante l'esperienza diretta con la realtà aziendale.

# INDICE

1	Introduzione .....	6
1.1	Obiettivi della tesi .....	7
1.2	Rassegna della letteratura .....	7
1.3	Contenuto della tesi .....	8
2	Presentazione del caso .....	10
2.1	La realtà aziendale .....	11
2.1.1	I rischi del mercato .....	12
2.1.2	L'uso della <i>Business Intelligence</i> .....	12
2.2	L' ERP OpenBravo .....	13
2.3	La base di dati operativa .....	14
3	Requisiti di analisi .....	18
3.1	Il Business aziendale .....	18
3.1.1	Le aziende cliente .....	20
3.1.2	L'offerta .....	20
3.2	Ordini di vendita .....	21
3.2.1	Raccolta dei requisiti degli Ordini di vendita .....	22
3.3	Venduto .....	22
3.3.1	Raccolta dei requisiti sul venduto .....	24
3.4	Campagne promozionali .....	24
3.4.1	Raccolta dei requisiti sulle Campagne promozionali .....	25
4	Progettazione dei Data Mart .....	27

4.1	Il processo di data warehousing .....	28
4.2	Il processo degli ordini di vendita .....	30
4.2.1	Specifica dei requisiti del fatto .....	30
4.2.2	Progettazione concettuale del <i>data mart</i> .....	34
4.3	Il processo del venduto .....	35
4.3.1	Specifica dei requisiti del fatto .....	36
4.3.2	Progettazione concettuale del <i>data mart</i> .....	39
4.4	Il processo delle campagne promozionali.....	39
4.4.1	Specifica dei requisiti del fatto .....	40
4.4.2	Progettazione concettuale del <i>data mart</i> .....	43
4.5	Tabelle riepilogative delle dimensioni e delle misure .....	43
5	Progettazione concettuale finale e logica dei data mart e del data warehouse	45
5.1	Analisi dei dati operazionali .....	46
5.2	Integrazione dei dati operazionali negli schemi concettuali dei <i>data mart</i> iniziali	48
5.3	Ordini di Vendita .....	49
5.3.1	Progettazione concettuale finale del <i>data mart</i> .....	49
5.3.2	Progettazione logica del <i>data mart</i> .....	50
5.4	Venduto e Campagne promozionali .....	53
5.4.1	Progettazione concettuale finale del <i>data mart</i> .....	53
5.4.2	Progettazione logica del data mart.....	53
5.5	Progettazione logica del <i>data warehouse</i> .....	54
6	Strumenti di sviluppo .....	56
6.1	Glossario.....	57

6.1.1	OLAP .....	57
6.1.2	MDX .....	58
6.1.3	PostgreSQL.....	60
6.2	Pentaho .....	61
6.2.1	La suite Pentaho Community Edition .....	61
6.3	pgAdmin .....	69
7	Procedure per la realizzazione dei <i>data mart</i> .....	70
7.1	La fase di estrazione.....	71
7.2	La fase di trasformazione .....	72
7.3	Il processo di caricamento .....	79
8	Il modello multidimensionale.....	82
8.1	La tecnologia OLAP.....	82
8.2	Progettazione dello schema OLAP .....	85
8.3	Definizione schema XML Mondrian .....	87
8.4	Pubblicazione sul server Pentaho .....	89
9	Definizione dei report.....	91
9.1	Analisi Ordini di vendita .....	92
9.2	Analisi Venduto .....	97
9.3	Analisi Campagne promozionali.....	104
9.4	Report.....	108
10	Conclusioni.....	112

## **I Bibliografia**

## **II Ringraziamenti**

# **1 INTRODUZIONE**

In questo capitolo viene mostrata una breve panoramica sul lavoro svolto, sulla letteratura a cui si è fatto riferimento e sulla struttura della tesi.

## 1.1 Obiettivi della tesi

Il presente lavoro di tesi è stato svolto presso l'azienda BNova S.r.l. di Massa, avente come partner la società Openia S.r.l. di Navacchio.

L'obiettivo generale della tesi è la progettazione e la realizzazione di strumenti analitici che permettano all'azienda committente (la Food & Beverage) di poter osservare il movimento del flusso monetario aziendale e la validità delle proprie campagne promozionali su un nuovo mercato.

In particolare, nella presente tesi si propone la progettazione e la realizzazione di un intero sistema di *Business Intelligence* utilizzando solo prodotti *open source*. Questo sistema permetterà di monitorare i tempi di saldo delle fatture da parte dei clienti, i tempi di consegna degli ordini e la validità delle campagne promozionali. Identificando così particolari problemi e/o rallentamenti che possono condurre alla creazione di sprechi nei processi aziendali.

Per raggiungere questo scopo i dati presenti nel gestionale dell'azienda sono stati organizzati in opportuni *database* (o *data mart*) progettati funzionalmente alle esigenze di analisi dei dati da parte degli utenti.

Per la parte di analisi e presentazione dei dati è stato utilizzato il sistema di *business intelligence open source Pentaho Community Edition*. In particolare utilizzando gli strumenti *On Line Analytical Processing* (OLAP) che questa suite mette a disposizione per questi scopi.

## 1.2 Rassegna della letteratura

Per l'analisi dei processi aziendali, ai fini della progettazione del *data warehouse*, si è fatto ampio riferimento al libro [Kim02]. In questo libro vengono affrontati una serie di problemi e date le relative soluzioni per la progettazione dei *data warehouse*.



Per la progettazione del *data warehouse* sono state consultate le soluzioni proposte da [Alb12]. Oltre che ad un'efficiente strutturazione del contenuto, sono state riscontrate alcune soluzioni adattabili al caso di studio aziendale. Le soluzioni di riferimento sono risultate molto efficaci grazie al corretto livello di astrazione che presentavano, ma anche alla possibilità di essere compresi dal committente del *data warehouse*, con i quali sono avvenuti frequenti scambi di idee e diversi momenti di verifica durante la fase di progettazione.

Anche la modulistica per la specifica dei requisiti ed il procedimento di progettazione sono quelli proposti da [Alb12].

Per le caratteristiche dei software e degli ambienti utilizzati per lo sviluppo del processo di *data warehousing*, e per le procedure di estrazione, trasformazione e caricamento si è fatto invece principalmente riferimento a [Bou09].

Durante il processo di creazione del cubo OLAP si fatto uso di [Rug12].

Nella creazione ed esplicitazione delle query MDX si fa riferimento a [Msd12].

## **1.3 Contenuto della tesi**

Il presente lavoro di tesi è organizzato nel modo seguente.

Nel Capitolo 2 viene presentata l'azienda committente, si fa riferimento alla sua struttura interna e del mercato in cui opera. Viene presentata la base di dati operativa e l'applicazione CRM da cui provengono i dati.

Nel Capitolo 3 vengono raccolti i requisiti di analisi e approfondita la struttura interna dell'azienda.

Nel Capitolo 4 viene progettata la prima versione dei *Data Mart*.

Nel Capitolo 5 viene descritta la versione finale dei *Data Mart* e la loro progettazione logica.

Nel Capitolo 6 vengono descritti gli strumenti di sviluppo utilizzati.

Nel Capitolo 7 vengono descritte le fasi di estrazione, trasformazione e caricamento dei dati.

Nel Capitolo 8 viene presentato il modello multidimensionale utilizzato.

Nel Capitolo 9 vengono definite le query MDX che rappresentano i report scaturiti dai requisiti di analisi.

Il Capitolo 10 presenta le conclusioni e i possibili sviluppi futuri.

## 2 PRESENTAZIONE DEL CASO

La maggior parte delle aziende è strutturata in dipartimenti o settori (finanziario, produttivo, controllo qualità, IT, risorse umane, commerciale, Marketing management) con proprie responsabilità, attività e obiettivi.

Ogni settore dispone di processi propri e di sistemi di *business*, che per potersi interfacciare (sia tra loro, che con il mondo esterno) e cooperare, devono osservare opportune direttive.

I vari dipartimenti, data la natura del loro operato e dei loro processi interni, hanno esigenze di analisi dei dati molto differenti tra loro; per gli scopi di questa tesi ci limiteremo quindi ad approfondire solo le specifiche relative al reparto commerciale e management coinvolti direttamente nella nostra analisi.

## 2.1 La realtà aziendale

La F&B (*Food & Beverage*) s.r.l. è un'azienda di medie dimensioni. Essa commercializza e distribuisce prodotti per il settore degli alimenti e delle bevande e conta. Ho un organico di 50 dipendenti e con sedi dislocate in Italia e all'estero. La F&B fa parte di una realtà aziendale dinamica e in continua espansione e ha realizzato nel 2010 un fatturato di circa 40 milioni di euro.

Nonostante un clima di generale recessione dell'industria italiana degli alimenti e delle bevande, la F&B ha confermato un'evoluzione tra le più dinamiche nel settore, mostrando una crescita del profitto di poco superiore all' 50% a prezzi costanti, che ha portato la società a espandersi rapidamente.

Tale crescita ha indotto l'azienda ad affacciarsi su nuovi mercati, ma, dato il periodo di crisi globale, tale espansione, seppur graduale, deve essere quanto mai controllata, selettiva e soprattutto stabile nel lungo periodo.

L'allargamento del business aziendale ha portato la F&B ad attuare un processo di ristrutturazione interna, passando da una struttura padronale (fortemente gerarchizzata) a una struttura manageriale (modello orizzontale) che ha comportato la riorganizzazione delle divisioni aziendali e l'ottimizzazione della gestione dei dati interni.

Per la gestione dei propri processi di business, l'azienda ha sinora optato per soluzioni informatiche *Open source*, quali il gestionale ERP OpenBravo.

La sigla ERP è l'acronimo di *Enterprise Resource Planning* (letteralmente "pianificazione delle risorse d'impresa"), ovvero un sistema gestionale, che integra tutti i processi di *business* rilevanti di un'azienda (vendite, acquisti, gestione magazzino, contabilità ecc.).

### 2.1.1 I rischi del mercato

La F&B opera in un mercato *B2B* (*Business-to-business*), in quanto ha clienti operanti nella GDO<sup>1</sup> e grossisti.

Per capire meglio i rischi del mercato in cui opera l'azienda bisogna introdurre un altro tipo di mercato esistente, quello *B2C* (*Business-to-consumer*), nel quale fanno parte i venditori che commercializzano al dettaglio.

Il volume di una transazione *B2B* è molto più elevato del volume di una transazione *B2C*. Questo comporta maggiori rischi nel momento dell'acquisto e della vendita di prodotti.

Come anticipato nel precedente paragrafo, l'azienda punta a espandersi su nuovi mercati. Questo porta a valutare una serie di rischi:

- *Merce non conforme*. Avere relazioni con nuove aziende può far nascere il rischio, nel breve periodo, di ricevere dai fornitori prodotti non in linea con gli standard aziendali (che dovranno essere rimpiazzati da altri), portando ad avere uno spreco di risorse e di denaro.
- *Prolungamento dei tempi di acquisto/vendita dei prodotti*. Dovuto al diverso modo di operare dei nuovi *partner* (fornitori, clienti).
- *Invenduto*. Il volume dei prodotti acquistati è molto superiore a quello che il mercato riesce ad assorbire, provocando l'accumulo di prodotti nei magazzini.

### 2.1.2 L'uso della *Business Intelligence*

L'espansione verso nuovi mercati porta l'azienda ad aumentare la propria competitività rendendo indispensabile un controllo accurato del modo di gestire le vendite e gli acquisti.

---

<sup>1</sup> È l'acronimo di Grande Distribuzione Organizzata e si riferisce al moderno sistema di vendita al dettaglio effettuato attraverso una rete di supermercati.

Nelle aziende più innovative si sta affermando una nuova concezione di *business*: i tradizionali modelli basati su prodotti e ricavi vengono sostituiti da modelli orientati ai clienti e ai profitti, i quali fanno entrare in gioco indicatori che permettono di massimizzare le potenzialità di pieno profitto per ogni cliente.

Per fruire di tali indicatori in modo rapido, semplice e a diversi livelli di dettaglio, l'azienda deve dotarsi di nuova tecnologia e formare manager in grado di comprendere le potenzialità e le possibilità dei nuovi modelli di analisi.

Un aiuto, in tal senso, può venire dagli strumenti di *Business Intelligence*, che permettono di monitorare costantemente l'operato di ogni settore aziendale in modo rapido e intuitivo, estrapolando informazioni consolidate e aggregate.

## 2.2 L' ERP OpenBravo

OpenBravo è un'applicazione ERP *Open source* e *web based*.

La tipica architettura di questo sistema è fondata su una componente *server*, dove risiedono i moduli che il gestionale espone agli utenti (Figura 2.1), e dalle postazioni client, che accedono all'applicazione tramite un normale *browser web*. OpenBravo permette di gestire le operazioni giornaliere, ed ottimizzare i processi sia aziendali che di *business*.

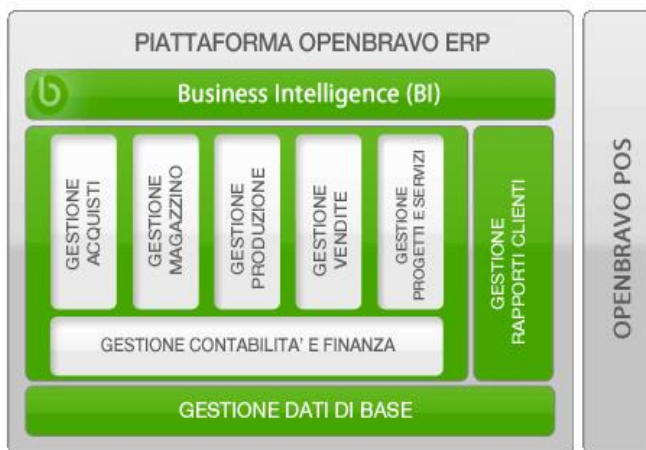


Figura 2.1 – La piattaforma OpenBravo

Le funzionalità principali di questo software sono:

- **Gestione dei dati principali** (prodotti, componenti, clienti, fornitori, agenti, impiegati, ecc...);

- **Gestione degli acquisti** (Ordini di acquisto, DDT di carico, fatture di acquisto, contabilità, ecc...);
- **Gestione del magazzino** (depositi, lotti, matricole, imballaggi, etichette, entrate, uscite, scorte, approvvigionamenti, trasporti);
- **Gestione dei progetti e dei servizi** (progetti, fasi, obiettivi, risorse, budget, spese, acquisti correlati);
- **Gestione della produzione** (pianificazione della produzione, ordini di produzione, report dei tempi, costi di produzione, manutenzione impianti);
- **Gestione delle vendite** (costi, imposte, ordini di vendita, spedizioni, commissioni, fatture);
- **Contabilità e Cespiti** (fatture, budget, tasse, depositi bancari, bilanci d'esercizio, immobilizzazioni tecniche).

## 2.3 La base di dati operativa

L'ERP OpenBravo utilizza un *database* operativa basato sul sistema PostgreSQL.

Il *database* presenta un gran numero di tabelle (circa 470) e di campi contenuti in esse (circa 30 per tabella). Per i nostri scopi sarà estratto un sotto-insieme (la parte significativa per le analisi) dello schema relazionale che contiene i dati inseriti a partire dal 01/01/2010, includendo anche tutte le informazioni collegate.

Il sotto insieme dello schema relazione è rappresentato nella Figura 2.2

Il primo campo di ogni tabella costituisce la chiave primaria e presenta la sigla PK prima del nome, i campi successivi costituiscono le chiavi esterne (sigla FK).

Tabelle appartenenti al gruppo Ordini:

- **C\_ORDER:** contiene i dati principali di un ordine. Ogni record è collegato all'agente che ha fatto e trasmesso l'ordine all'azienda.





- **C\_INVOICE:** contiene i dati globali delle fatture e può essere chiamata anche tabella di testata. Ogni record specifica la data di emissione della fattura (*dateinvoiced*) e il totale (*grandtotal*) ed è collegata ad un cliente, un rappresentante, una succursale, ai termini di pagamento e a una campagna di marketing.
- **C\_INVOICELINE:** contiene i dati delle righe di ogni fattura. Ogni record specifica il prezzo del prodotto (*priceactual*), la quantità fatturata (*qtyinvoiced*), il ricavo totale netto (*linenetamt*) e la quantità di tasse (*taxamt*) ed è collegato a una fattura e un prodotto.
- **C\_PAYMENTTERM:** contiene la lista dei termini di pagamento.
- **FIN\_PAYMENT:** contiene la lista dei pagamenti effettuati da un cliente in riferimento a una fattura. Ogni record contiene la data in cui è stato effettuato il pagamento (*paymentdate*) e l'ammontare pagato (*amount*).

Tabelle appartenenti al gruppo sedi:

- **AD\_CLIENT:** contiene la lista delle sedi "Succursale". Ogni record specifica la ragione sociale (*Name*) ed è collegato a una sede "Capogruppo".
- **AD\_ORG:** contiene la lista delle sedi "Capogruppo". Ogni record specifica la ragione sociale (*Name*) ed è collegato ai dettagli di una sede "Capogruppo".
- **AD\_ORG\_INFO:** contiene la lista dei dettagli delle sedi "Capogruppo". Ogni record è collegato alla lista degli indirizzi. Una sede "Capogruppo" può avere più indirizzi.

Tabelle appartenenti al gruppo entità:

- **C\_BPARTNER:** contiene la lista delle entità con cui l'azienda ha un'iterazione (da ora in poi *partner*). Queste entità possono essere i rappresentanti (quando il valore del campo *ISsaresrep* è Y) e/o gli impiegati (quando il valore del campo *ISemployee* è Y) e/o i clienti (quando il valore del campo *IScustomer* è Y). Inoltre ogni record specifica il nome o la ragione sociale (*Name*), il fido massimo (*so\_creditused*) e il fido utilizzato (*so\_creditlimit*).

- **C\_BPARTNER\_LOCATION:** definisce l'indirizzo fisico di un *partner*. Un *partner* può avere più indirizzi.
- **C\_LOCATION:** contiene la lista degli indirizzi. Ogni record specifica il nome della città (*City*) ed è collegato con una regione.
- **C\_REGION:** contiene la lista delle regioni. Ogni record specifica il nome della regione (*Name*).

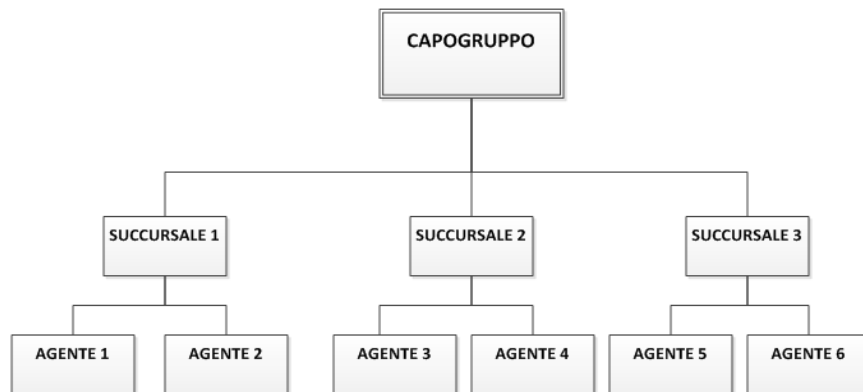
Tabelle appartenenti al gruppo prodotti e campagne promozionali:

- **M\_PRODUCT:** contiene la lista dei prodotti con cui l'azienda ha avuto un'interazione (sono inclusi anche i servizi). Ogni record specifica se il prodotto è stato venduto (quando il valore di *ISold* è uguale a Y), la sigla (*Value*), la descrizione (*Name*), il prezzo di acquisto (*Cost*) e le unità presenti in ogni imballo (*Unitperpallet*). Ogni prodotto è collegato a una categoria.
- **M\_PRODUCT\_CATEGORY:** contiene la lista delle categorie dei prodotti. Ogni record specifica il nome della categoria (*Name*) e la macro categoria di appartenenza (*producttype*).
- **C\_CAMPAIGN:** contiene i dati delle campagne di marketing. Ogni record specifica la data di inizio e fine, il costo e il nome di ogni campagna.

## 3 REQUISITI DI ANALISI

### 3.1 Il Business aziendale

La F&B possiede sedi dislocate in molti Paesi. Queste sedi hanno una struttura gerarchica rappresentata in Figura 3-1.



**Figura 3-1 – Struttura delle sedi**

Al livello superiore si trova la sede chiamata “Capogruppo” che propone alle altre sedi gli obiettivi da raggiungere.

Al livello inferiore troviamo le sedi chiamate “Succursale” che rispondono alla sede “Capogruppo”. La sede “Capogruppo” è a sua volta una sede “Succursale”, in

quanto è operativa sul mercato. A ogni sede “Succursale” è associato un magazzino per i prodotti.

Al livello più basso troviamo gli agenti. A ogni agente è associata una e una sola “Succursale”, che può cambiare nel tempo. Ogni agente ha un'area di competenza nella quale opera e risponde direttamente a un proprio responsabile di vendita.

L'azienda nel corso degli anni ha sviluppato il suo processo di *business* secondo il seguente schema:

1. **Acquisto di prodotti per l'approvvigionamento iniziale.** I prodotti vengono acquistati dalle aziende fornitrici, che possono essere direttamente le aziende produttrici del prodotto o aziende intermediarie.
2. **Controllo qualitativo e quantitativo del materiale ricevuto.** I prodotti ricevuti vengono controllati per evitare che ci sia materiale non conforme alle direttive aziendali.
3. **Immagazzinamento.** I prodotti vengono riposti nei magazzini delle sedi “Succursale”.
4. **Ricezione ordine.** Viene ricevuto un ordine di merci da parte di un'azienda cliente.
5. **Evasione dell'ordine.** L'ordine viene evaso e la merce spedita all'azienda cliente.
6. **Controllo della giacenza della merce nel magazzino.** Viene controllato se nel magazzino la giacenza della merce non sia scesa sotto una certa soglia per effetto dell'ordine ricevuto e in questo caso si procede al riapprovvigionamento dei prodotti.
7. **Fatturazione.** Dopo la ricezione della merce da parte dell'azienda cliente, la “Succursale” invia la fattura.
8. **Venduto.** Al momento della ricezione del pagamento della fattura, la merce presente in quella fattura viene segnalata come venduta.

### 3.1.1 Le aziende cliente

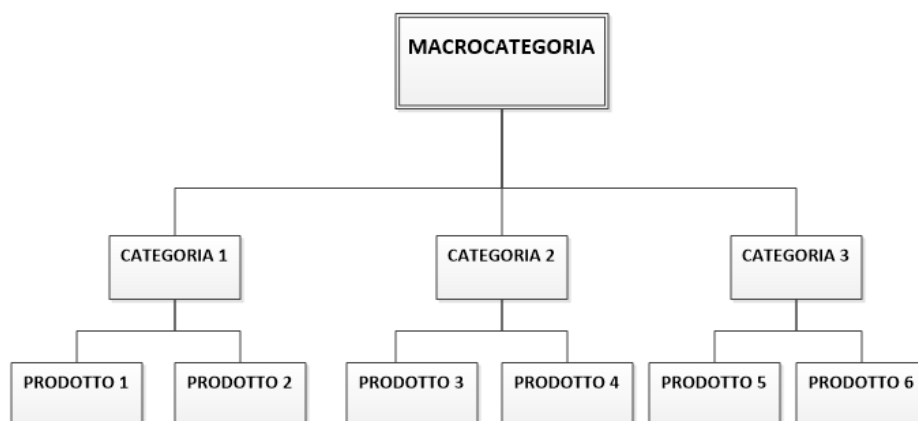
Le aziende cliente sono principalmente catene (hotel e ristoranti) e aziende della Grande Distribuzione Organizzata (GDO)<sup>2</sup>.

Ad ogni cliente viene proposto uno specifico contratto basato su vari fattori quali lo storico e/o il previsionale del volume degli ordini, la puntualità e la regolarità dei pagamenti (per clienti già fidelizzati), ed in base a questi fattori viene anche definito un piano di scontistiche personalizzate.

### 3.1.2 L'offerta

L'azienda non ha una catena di produzione di prodotti propri, ma li acquista dai fornitori basandosi sugli ordini effettuati dalle aziende cliente, di conseguenza ogni "Succursale" si trova a gestire gamme diverse di prodotti.

L'azienda acquista e vende prodotti, quasi tutti con *brand* molto conosciuti, nel settore alimentare, sia freschi (come latte, uova, formaggi), sia a lunga scadenza (come prodotti in scatola), bevande alcoliche (come liquori e birre) e analcoliche (come bevande gassate, succhi di frutta).



**Figura 3-2 – Struttura gerarchica delle categorie di prodotti**

---

<sup>2</sup> Con il termine Grande Distribuzione Organizzata ci si riferisce al moderno sistema di vendita al dettaglio effettuato attraverso una rete di supermercati.

Ogni prodotto può essere venduto singolarmente, oppure in confezioni che possono contenere da due fino a un massimo di sei prodotti.

I prodotti, dal punto di vista del business aziendale, sono divisi in categorie e macro categorie. Ogni prodotto può avere una sola categoria e ogni categoria può avere una sola macro categoria, come illustrato in Figura 3-2.

## **3.2 Ordini di vendita**

Per ordine di vendita si intende, dal punto di vista del venditore, l'intenzione di vendere al cliente i prodotti che quest'ultimo ha ordinato tramite un ordine di acquisto.

L'agente è il responsabile delle attività che riguardano gli ordini di vendita e l'interazione con il cliente.

A intervalli di tempo prestabiliti (circa due mesi) vengono effettuate delle verifiche sul totale degli ordini ricevuti rispetto a quelli preventivati a budget in modo da poter intraprendere azioni correttive e/o promozionali per incrementarne il volume.

Quando viene effettuato un ordine di acquisto da parte di un'azienda cliente, questo viene formalizzato dall'agente e trasformato in ordine di vendita. Esso viene inserito nel sistema gestionale, dove viene specificato, per ogni riga d'ordine, una o più date di consegna prevista, ovvero la data nella quale ci si aspetta che la sede "Succursale" consegna la merce all'azienda cliente.

La data di consegna prevista è molto importante perché l'intero processo di vendita è basato sulla velocità di approvvigionamento dei prodotti. Una maggiore velocità di approvvigionamento porta una diminuzione del tempo di consegna delle merci.

Un ordine di vendita riguarda una sola azienda cliente, un solo agente di vendita e una sola sede "Succursale". A ogni riga d'ordine di vendita corrisponde un solo

prodotto con la quantità ordinata. Lo stesso prodotto può essere presente in più righe.

Monitorare il venduto è importante, sia per verificare il rispetto delle quantità definite a budget, sia per definire un *trend* di vendita più significativo rispetto a quello previsto ad inizio esercizio. Da questo monitoraggio ne trarrà beneficio anche il settore degli approvvigionamenti, perché avrà a disposizione una proiezione degli acquisti futuri in base alla vendita dei prodotti nel presente.

### 3.2.1 Raccolta dei requisiti degli Ordini di vendita

Il settore commerciale è interessato a effettuare le seguenti analisi:

#### *Requisiti di analisi del processo Ordini di vendita*

1	Ordinato totale per agente e sede "Succursale".
2	Rango dei clienti e delle "aree geografiche" in termini di volume degli ordini nell'ultimo anno.
3	I 10 prodotti ordinati in un certo mese con la quantità più alta, per categoria di prodotto.
4	Tempi medi di attesa per evasione degli ordini.

Per semplificare la scrittura dei requisiti di analisi, oltre a restare valide le considerazioni fatte in precedenza, si è supposto che:

- Con l'analisi per prodotto si intende la necessità di analizzare anche per macro categoria, categoria e codice prodotto.
- Con l'analisi per cliente si intende la necessità di analizzare anche per area geografica, città e ragione sociale cliente.

## 3.3 Venduto

La sede "Succursale", dopo aver ricevuto dal cliente l'accettazione della merce ordinata, provvede all'invio la fattura relativa all'ordine di vendita. Per via della grande quantità di merce presente nell'ordine di vendita, vengono associate una o più fatture.

Ad ogni fattura, anche se fa riferimento allo stesso ordine, viene associata una modalità di pagamento e la possibilità di pagamenti parziali.

I tipi di pagamento possono essere cinque:

- a) **Pagamento alla ricezione della fattura.** Il pagamento deve avvenire entro un giorno feriale dalla data di ricezione della fattura.
- b) **Pagamento a 15 giorni.** Il pagamento deve avvenire entro 15 giorni dalla data di ricezione della fattura. Se il quindicesimo giorno è festivo, allora dovrà avvenire il primo giorno feriale utile.
- c) **Pagamento a 30 giorni.** Il pagamento deve avvenire entro 30 giorni dalla data di ricezione della fattura. Se il trentesimo giorno è festivo, allora dovrà avvenire il primo giorno feriale utile.
- d) **Pagamento a 60 giorni.** Il pagamento deve avvenire entro 60 giorni dalla data di ricezione della fattura. Se il trentesimo giorno è festivo, allora dovrà avvenire il primo giorno feriale utile.
- e) **Pagamento a 90 giorni.** Il pagamento deve avvenire entro 90 giorni dalla data di ricezione della fattura. Se il novantesimo giorno è festivo, allora dovrà avvenire il primo giorno feriale utile.

L'informazione che rappresenta la quantità di giorni intercorsi tra l'emissione della fattura e il suo pagamento, è molto importante ai fini delle analisi aziendali, in quanto permette di:

- 1) **Individuare i tempi medi di pagamento delle fatture.** In modo da migliorare la liquidità riducendo i tempi di pagamento per i clienti più puntuali e aumentandoli per i clienti più ritardatari.
- 2) **Sollecitare pagamento di fatture in scadenza o appena scaduta.**
- 3) **Individuare quando una fattura sta per diventare insoluta.** Anticipando le procedure per avviare eventualmente le pratiche legali.
- 4) **Semplificare il lavoro del commerciale.** La divisione commerciale sarà facilitata nell'individuare fatture il cui pagamento andrà a buon fine.



L'azienda F&B permette a tutte le aziende cliente di avere un fido, ha quindi necessità di tenere sotto controllo l'ammontare del fido utilizzato ed essere avvisata quando un'azienda cliente supera la soglia massima.

### 3.3.1 Raccolta dei requisiti sul venduto

Il settore commerciale è interessato a effettuare le seguenti analisi:

*Requisiti di analisi del processo Venduto*

5	Fatturato totale per agente e sede "Succursale".
6	Rango degli agenti in termini di volume del venduto per mese.
7	I 10 prodotti venduti in un certo mese con il volume di vendita più alto, per categoria di prodotto.
8	Tempi attesa medi di saldo delle fatture.
9	Marginalità delle vendite rispetto ai costi dei prodotti.

Per semplificare la scrittura dei requisiti di analisi, oltre a restare valide le considerazioni fatte in precedenza, si è supposto che con l'analisi per agente si intende la necessità di analizzare anche per sede "Capogruppo" e sede "Succursale".

## 3.4 Campagne promozionali

Un'azienda non può espandersi verso nuovi mercati senza un'adeguata campagna promozionale di norma gestita dal *Marketing management*.

Tale divisione ha il compito di utilizzare la combinazione di informazioni provenienti dal *know-how* aziendale e informazioni esterne all'azienda, sia in riferimento al nuovo target di clienti, sia in riferimento al portafoglio clienti per aumentare la loro fidelizzazione.

Le campagne promozionali utilizzate dal *Marketing management* aziendale si possono racchiudere in due grandi gruppi:

- **Applicando sconti quantità.** Ha l'obiettivo di premiare le aziende clienti che acquistano quantitativi elevati di prodotti applicando uno sconto in base alla

quantità richiesta. La percentuale di sconto può variare in base all'azienda cliente oppure in base all'area geografica.

- **Con omaggi.** Ha come obiettivo l'aumento di fidelizzazione di un'azienda cliente, elargendo omaggi in base alla quantità di merce acquistata con offerte nella forma "prendi tre paghi due".

Relativamente alle informazioni interne all'azienda, il *Marketing manager* giudica l'andamento di una campagna promozionale in base ai seguenti fattori:

- A. Se un prodotto in promozione aumenta le vendite durante il periodo promozionale. I valori di base sono stimati a priori in base all'analisi delle vendite trascorse.
- B. Se un prodotto in promozione mostra un calo di vendite dopo il periodo di promozione, comportando il totale o parziale annullamento dei profitti generati con la promozione (questo effetto si chiama *time shifting*).
- C. Se un prodotto in promozione mostra un aumento di vendite ma gli altri prodotti omogenei presentano una diminuzione (questo effetto si chiama *cannibalizzazione*).
- D. Se tutti i prodotti della categoria in promozione hanno un aumento netto di vendite sia prima, sia durante e dopo la promozione (questo effetto si chiama *crescita di mercato*).
- E. Se la promozione ha portato un profitto. Il profitto di una promozione è dato dal guadagno calcolando il profitto scaturito dall'incremento delle vendite della categoria di prodotti in promozione, rimuovendo i costi, se presenti, della *cannibalizzazione*, del *time shifting* e quelli relativi alla promozione (vengono inclusi i costi per la diminuzione di prezzo dei prodotti e della campagna pubblicitaria).

### 3.4.1 Raccolta dei requisiti sulle Campagne promozionali

Il settore del *Marketing management* è interessato a effettuare le seguenti analisi:

*Requisiti di analisi del processo Campagne promozionali*

---

10	Ricavato da campagne promozionali per sede "Capogruppo".
11	Nuovi clienti derivati da campagne per sede "Capogruppo".
12	Confronto andamento campagna rispetto al budget preventivato.
13	Analisi incremento vendite per prodotto e per cliente.

Per semplificare la scrittura dei requisiti di analisi restano valide le considerazioni fatte in precedenza.

## 4 PROGETTAZIONE DEI DATA MART

Dopo una breve introduzione al processo di *data warehousing*, si documenta formalmente la progettazione concettuale del *data warehouse* a partire dalle analisi dei requisiti.

Viene posto il focus sulle specifiche dei requisiti e sul disegno concettuale iniziale dei *data mart* sulla base dei requisiti di analisi.

## 4.1 Il processo di data warehousing

Con il termine *data warehouse* [Inm96][KR02] si fa riferimento a una collezione di dati avente le seguenti caratteristiche:

- **Tempificata.** Contiene informazioni sul tempo in cui si verificano gli eventi di interesse. Questa caratteristica permette di realizzare la storicizzazione delle informazioni, tramite fotografie e variazioni.
- **Integrata.** I dati possono provenire da sorgenti eterogenee.
- **Statica.** I dati sono utilizzati solo per operazioni di ricerca e non di modifica.
- **Organizzata per soggetti.** I dati sono organizzati per analizzare soggetti di interesse.
- **Finalizzata ad analisi di supporto alle decisioni.** I dati sono organizzati per facilitare la loro analisi al fine di produrre opportune sintesi.

Il termine *data warehousing* si riferisce al processo eseguito per organizzare e caricare i dati estratti dalla base di dati operativa in un *data warehouse* e per consentire agli utenti di analizzarli con strumenti di *Business Intelligence*, termine che indica la capacità di utilizzare l'informazione prodotta da sistemi informativi eterogenei ed interpretarli in modo intelligente con lo scopo di migliorare ed ottimizzare i flussi informativi ed i processi.

Fra le tre possibili soluzioni architetturali adottabili, le quali variano a seconda del numero di livelli di memorizzazione dei dati, si è optato per l'utilizzazione di una *architettura a due livelli* la quale prevede il caricamento diretto del *data warehouse*

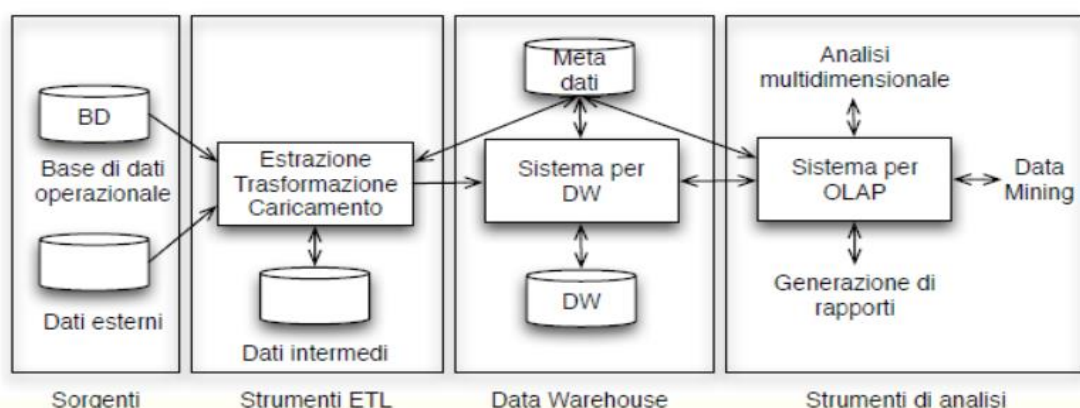


Figura 4-1 - Architettura a tre livelli del processo di *data warehousing* [Alb12]

a partire dai dati operazionali opportunamente modificati.

Come mostrato in Figura 4-1, il processo di *data warehousing* è composto di quattro fasi principali:

- **Sorgenti.** Definizione delle sorgenti dei dati, base di dati operazionale ed eventualmente fonti esterne.
- **Extraction Trasformation Loading.** Fase in cui i dati provenienti dalle sorgenti vengono opportunamente estratti, modificati e caricati nel sistema per *data warehouse*.
- **Data warehouse.** Sistema in cui sono opportunamente memorizzati i dati con lo scopo di ottimizzare la loro successiva analisi.
- **Strumenti di analisi.** Insieme di strumenti che interrogando il *data warehouse* permettono di creare reportistica *ad-hoc*, analizzare i dati in modo multidimensionale ed effettuare operazioni di *data mining*.

Una volta creato e popolato, il *data warehouse* necessiterà di operazioni di aggiornamento effettuate ad intervalli di tempo regolari per garantire l'allineamento con la base di dati operazionale.

Un *data mart* è un particolare sotto insieme del *data warehouse* contenente i dati relativi ad un particolare settore o processo aziendale. All'interno di una organizzazione possono quindi esistere più *data mart*, aventi finalità diverse e orientati a coprire le diverse aree di business. Le caratteristiche principali del *data mart* sono:

- Modellazione completamente guidata dalle analisi da effettuare e dal livello di performance.
- Periodicità di caricamento funzionale al *reporting* da produrre, compatibilmente con la frequenza di aggiornamento delle informazioni sottostanti nel *data warehouse*.

## 4.2 Il processo degli ordini di vendita

Si descrive la specifica dei requisiti e la progettazione concettuale iniziale del *data mart* Ordini di vendita.

L'ordine di vendita rappresenta una richiesta di prodotti da parte di un'azienda cliente. Ciascun ordine di vendita è formato da tante righe ognuna delle quali riguarda un prodotto. Un prodotto può trovarsi in più righe in funzione di differenti configurazioni di imballaggio e/o condizioni di vendita.

### 4.2.1 Specifica dei requisiti del fatto

Dai requisiti di analisi raccolti in precedenza (Paragrafo 3.2.1) discende la seguente specifica dei requisiti del fatto Ordini di vendita che associa a ogni requisito di analisi le dimensioni, le misure e le aggregazioni coinvolte.

<i>Specifica dei requisiti del fatto Ordini di vendita</i>			
<b>N.</b>	<b>Requisito di analisi</b>	<b>Dimensioni</b>	<b>Misure</b>
1	Ordinato totale per agente e sede "Succursale".	Agente, Sede "Succursale", NumeroOrdine	Quantità, prezzoUnitario, importoNetto, tasse, importoLordo
2	Rango dei clienti e delle "aree geografiche" in termini di volume degli ordini nell'ultimo anno.	Cliente, NumeroOrdine	Quantità, importoNetto
3	I 10 prodotti ordinati in un certo mese con la quantità più alta, per categoria di prodotto.	Prodotto, Data, NumeroOrdine	Quantità
4	Tempi medi di attesa per evasione degli ordini in un certo mese e per cliente	Data, Cliente, NumeroOrdine	GiorniEvasione

Si descrive il fatto Ordini di vendita, specificandone la granularità, e quindi il suo significato, le dimensioni e le misure preliminari interessate.

<i>Dimensioni preliminari del fatto Ordini di vendita</i>		
<b>Descrizione</b>	<b>Dimensioni preliminari</b>	<b>Misure preliminari</b>
La riga dell'ordine di vendita. Ad ogni riga corrisponde un prodotto ordinato da un cliente	Agente, Succursale, Prodotto, Cliente, Data, NumeroOrdine	Quantità, PrezzoUnitario, importoNetto, tasse, importoLordo

Si descrivono le dimensioni del fatto Ordini di vendita specificando per ognuna di esse il nome, una descrizione e la granularità.

*Dimensioni - Riepilogo*

<b>Nome</b>	<b>Descrizione</b>	<b>Granularità</b>
Agente	L'agente da cui è partito l'ordine	Un agente
Cliente	L'azienda cliente che ordina il prodotto	Un cliente
Data	Dimensione temporale in cui si svolgono i fatti	Un giorno
Succursale	La sede "Succursale" a cui fa capo l'agente	Una sede
Prodotto	Il prodotto finito che viene spedito a seguito di un ordine	Un prodotto
NumeroOrdine	Il numero dell'ordine di vendita che raggruppa più righe	Un ordine di vendita

Per ogni dimensione vengono elencati gli attributi e una loro descrizione.

*Dimensione – Agente*

<b>Attributo</b>	<b>Descrizione</b>
NomeCognomeAgente	Nome e cognome dell'agente
CittaAgente	Città dell'agente
AreaGeograficaAgente	Area geografica dell'agente

*Dimensione - Cliente*

<b>Attributo</b>	<b>Descrizione</b>
RagioneSocialeCliente	Ragione sociale del cliente
CittaCliente	Città del cliente
AreaGeograficaCliente	Area geografica del cliente

*Dimensione - Data*

<b>Attributo</b>	<b>Descrizione</b>
Anno	Anno
Semestre	Semestre
Trimestre	Trimestre
Mese	Mese
Data	Data

*Dimensione - Succursale*

<b>Attributo</b>	<b>Descrizione</b>
RagioneSocialeSuccursale	Ragione sociale della sede
Capogruppo	Ragione sociale della sede "Capogruppo"
CittaSuccursale	Città della succursale
AreaGeograficaSuccursale	Area geografica della succursale



*Dimensione - Prodotto*

Attributo	Descrizione
SiglaProdotto	Sigla del prodotto
NumeroProdottImballo	Quantità di prodotto contenuto in una confezione
MacroCategoriaProdotto	Nome della macro categoria del prodotto
CategoriaProdotto	Nome della categoria del prodotto

*Dimensione - NumeroOrdine*

Attributo	Descrizione
NumeroOrdine	Numero dell'ordine di vendita

Si descrivono le gerarchie dimensionali specificando per ogni dimensione le possibili gerarchie fra gli attributi e il loro tipo (bilanciata, incompleta, ricorsiva).

*Gerarchie dimensionali*

Dimensione	Descrizione gerarchia	Tipo gerarchia
Data	Data-> Mese-> Trimestre-> Semestre-> Anno	Bilanciata
Cliente	Cliente -> CittaCliente -> AreaGeograficaCliente	Bilanciata
Succursale	RagioneSocialeSuccursale-> CittaSuccursale -> AreaGeograficaSuccursale	Bilanciata
Agente	NomeCognomeAgente -> RagioneSocialeSuccursale-> CittaSuccursale -> AreaGeograficaSuccursale	Bilanciata
Prodotto	SiglaProdotto-> CategoriaProdotto -> MacroCategoriaProdotto	Bilanciata

Si specifica il tipo di strategia da usare per trattare le dimensioni con attributi che possono cambiare nel tempo. Si elenca quindi la lista delle dimensioni ricavate, evidenziandone la granularità e la metodologia da utilizzare per il trattamento delle modifiche.

Le tipologie di trattamento delle modifiche possibili sono quattro, delle quali le prime tre si considerano quando qualche attributo dimensionale cambia raramente (*slowly changing dimensions*) [Alb12]:

- 1) **Perdita di storia (Tipo 1).** Il valore dell'attributo dimensionale che cambia viene sostituito con il nuovo valore. È la soluzione più semplice e immediata, ma si perde la possibilità di storicizzare i cambiamenti.
- 2) **Storicizzazione (Tipo 2).** Si aggiunge una nuova riga alla tabella dimensionale, creando di fatto una entità nuova. Tutti i fatti precedenti alla

modifica fanno riferimento alla vecchia entità, mentre tutti i fatti successivi alla modifica fanno riferimento a quella nuova. In questo modo il caricamento dei dati si complica, aumentano i dati della dimensione, ma si ha la possibilità di storicizzare il cambiamento.

- 3) **Storicizzazione con data cambiamento (Tipo 3).** Con questo trattamento si memorizza, oltre che alla storia, anche il momento temporale in cui avviene il cambiamento. Per ottenere questo risultato è necessario sostituire l'Attributo con tre campi: Attributo, NuovoAttributo, DataModifica.
- 4) **Elevata frequenza di cambiamento (Tipo 4).** Per gli attributi dimensionali che cambiano molto frequentemente si possono prevedere due tabelle dimensionali, una contenente gli attributi che rimangono immutati e una contenente gli attributi che variano.

Di seguito viene descritta la modalità di trattamento delle dimensioni coinvolte nel fatto Ordini di vendita.

<i>Dimensioni</i>			
Nome	Granularità	Parzialità	Trattamento modifiche
Agente	Un agente	NO	Tipo 2
Cliente	Un cliente	NO	Tipo 2
Data	Una data	NO	Tipo 1
Succursale	Una sede "Succursale"	NO	Tipo 2
Prodotto	Un prodotto	NO	Tipo 2
NumeroOrdine	Un ordine di vendita	NO	Tipo 1

Si descrive ogni misura del fatto Ordini di vendita, come derivarla da altre misure, se calcolata, e il tipo di aggregabilità.

<i>Misure</i>			
Misura	Descrizione	Aggregabilità	Derivata
ImportoNetto	Quantità * PrezzoUnitario	Additiva	SI
PrezzoUnitario		Non additiva	NO
Quantità		Additiva	NO
Imposta	ImportoNetto * Tasse[nota %] /100	Additiva	SI
ImportoLordo	ImportoNetto + Imposta	Additiva	SI
GiorniEvasione	Tempo di attesa di evasione di un ordine	Semi additiva	SI

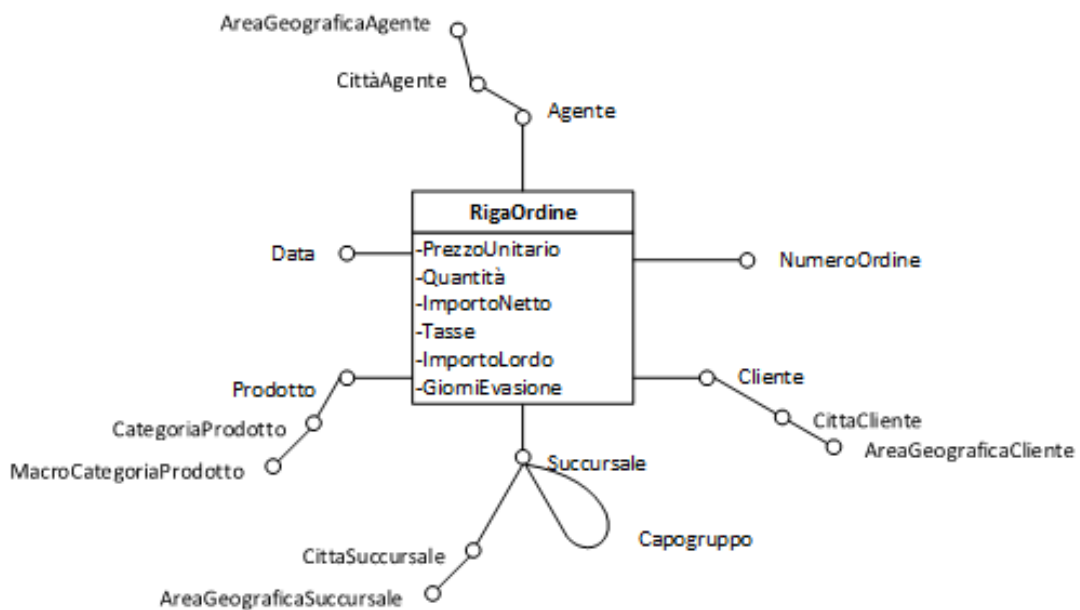
### 4.2.2 Progettazione concettuale del *data mart*

Partendo dalle definizioni formali delle analisi dei requisiti, viene definito un modello concettuale dei dati da analizzare, rappresentato mediante il *Dimensional Fact Model* o DFM [MGR98] che prevede la creazione di uno schema di fatto per l'oggetto di analisi.

Tale schema è in grado di descrivere graficamente tutti i concetti del modello multidimensionale, con l'utilizzo di sfumature concettuali per caratterizzare gli scenari più complessi. Il DFM si pone i seguenti obiettivi:

- creare un ambiente in cui le interrogazioni utente possano essere formulate in modo intuitivo;
- rendere possibile la comunicazione tra il progettista e l'utente finale con l'obiettivo di raffinare le specifiche dei requisiti;
- costituire un modello di partenza per la progettazione logica.

Tale modello è indipendente dal modello logico che sarà adottato nella successiva fase di modellazione logica.



**Figura 4-2 – Schema concettuale iniziale del *data mart* Ordini di vendita**

Lo schema concettuale del *data mart* Ordini di vendita è presentato in Figura 4-2, nella quale è possibile notare la presenza di:

- A. *un fatto*: gli ordini di vendita;
- B. *sei misure*: ImportoNetto, PrezzoUnitario, Quantità, Tasse, ImportoLordo, GiorniEvasione;
- C. *quattro gerarchie dimensionali*: Cliente, Data, Succursale, Prodotto;
- D. *una dimensione degenera*: NumeroOrdine, che permette di effettuare analisi con diverso livello di dettaglio, facendo in modo di poter analizzare il fatto Ordini di vendita, oltre che per righe, anche per i singoli ordini.

## 4.3 Il processo del venduto

Si descrive la specifica dei requisiti e la progettazione concettuale iniziale del *data mart* Venduto.

Un venduto rappresenta una fattura emessa da parte di una sede “Succursale” a un’ azienda cliente. Ciascuna fattura comprende tante righe ognuna riguardante un prodotto. Un prodotto può trovarsi in più righe in funzione di differenti configurazioni di imballaggio e/o condizioni di vendita.

### 4.3.1 Specifica dei requisiti del fatto

Dai requisiti di analisi raccolti in precedenza (Paragrafo 3.3.1) discende la seguente specifica dei requisiti del fatto Fatture che associa a ogni requisito di analisi le dimensioni, le misure e le aggregazioni coinvolte.

*Specifica dei requisiti del fatto Venduto*

N.	Requisito di analisi	Dimensioni	Misure
5	Fatturato totale per agente e sede "Succursale".	Agente, Sede "Succursale", NumeroFattura	importoNetto
6	Rango degli agenti in termini di volume del venduto per mese.	Cliente, Agente, NumeroFattura	importoNetto
7	I 10 prodotti venduti in un certo mese con il volume di vendita più alto, per categoria di prodotto.	Prodotto, Data, NumeroFattura	Quantità
8	Tempi attesa medi di saldo delle fatture.	Data, NumeroFattura	giorniPagamento
9	Marginalità delle vendite rispetto ai costi dei prodotti.	Prodotto, NumeroFattura	prezzoUnitario

Si descrive il fatto Fatture, specificandone la granularità, e quindi il suo significato, le dimensioni e le misure preliminari interessate.

La granularità del singolo fatto determina la dimensione del *data mart* e il tipo di analisi che si può effettuare sui dati; partendo da un dato ad elevata granularità è possibile ottenere altri dati meno dettagliati.

*Dimensioni preliminari del fatto Venduto*

Descrizione	Dimensioni preliminari	Misure preliminari
La riga della fattura. Ad ogni riga corrisponde un prodotto venduto ad un cliente	Agente, Succursale, Prodotto, Cliente, Data, NumeroFattura	Quantità, importoNetto, prezzoUnitario, GiorniPagamento

Si descrivono le dimensioni del fatto Fatture specificando per ognuna di esse il nome, una descrizione e la granularità.

*Dimensioni - Riepilogo*

Nome	Descrizione	Granularità
Agente	L'agente da cui è partito l'ordine	Un agente
Cliente	L'azienda cliente che ordina il prodotto	Un cliente
Data	Dimensione temporale in cui si svolgono i fatti	Un giorno
Succursale	La sede "Succursale" a cui fa capo l'agente	Una sede

Prodotto	Il prodotto finito che viene spedito a seguito di un ordine	Un prodotto
NumeroFattura	Il numero della fattura che raggruppa più righe	Una fattura

Per ogni dimensione vengono elencati gli attributi e una loro descrizione.

*Dimensione - Agente*

Attributo	Descrizione
NomeCognomeAgente	Nome e cognome dell'agente
CittaAgente	Città dell'agente
AreaGeograficaAgente	Area geografica dell'agente

*Dimensione - Cliente*

Attributo	Descrizione
RagioneSocialeCliente	Ragione sociale del cliente
FidoMassimo	Fido massimo del cliente
CittaCliente	Città del cliente
AreaGeograficaCliente	Area geografica del cliente

*Dimensione - Data*

Attributo	Descrizione
Anno	Anno
Semestre	Semestre
Trimestre	Trimestre
Mese	Mese
Data	Data

*Dimensione - Succursale*

Attributo	Descrizione
RagioneSocialeSuccursale	Ragione sociale della sede
Capogruppo	Ragione sociale della sede "Capogruppo"
CittaSuccursale	Città della succursale
AreaGeograficaSuccursale	Area geografica della succursale

*Dimensione - Prodotto*

Attributo	Descrizione
SiglaProdotto	Sigla del prodotto
Costo	Costo del prodotto
NumeroProdottimballo	Quantità di prodotto contenuto in una confezione
MacroCategoriaProdotto	Nome della macro categoria del prodotto
CategoriaProdotto	Nome della categoria del prodotto

*Dimensione - NumeroFattura*

Attributo	Descrizione
NumeroFattura	Numero della fattura

Si descrivono le gerarchie dimensionali specificando per ogni dimensione le possibili gerarchie fra gli attributi e il loro tipo (bilanciata, incompleta, ricorsiva).

<i>Gerarchie dimensionali</i>		
Dimensione	Descrizione gerarchia	Tipo gerarchia
Data	Data-> Mese-> Trimestre-> Semestre-> Anno	Bilanciata
Cliente	Cliente -> CittaCliente -> AreaGeograficaCliente	Bilanciata
Succursale	RagioneSocialeSuccursale-> CittaSuccursale -> AreaGeograficaSuccursale	Bilanciata
Agente	NomeCognomeAgente -> RagioneSocialeSuccursale-> CittaSuccursale -> AreaGeograficaSuccursale	Bilanciata
Prodotto	SiglaProdotto-> CategoriaProdotto -> MacroCategoriaProdotto	Bilanciata

Di seguito viene descritta la modalità di trattamento delle dimensioni coinvolte nel fatto Fatture.

<i>Dimensioni</i>			
Nome	Granularità	Parzialità	Trattamento modifiche
Agente	Un agente	NO	Tipo 2
Cliente	Un cliente	NO	Tipo 2
Data	Una data	NO	Tipo 1
Succursale	Una sede "Succursale"	NO	Tipo 2
Prodotto	Un prodotto	NO	Tipo 2
NumeroFattura	Una fattura	NO	Tipo 1

Si descrive ogni misura del fatto Ordini di vendita, come derivarla da altre misure, se calcolata, e il tipo di aggregabilità.

<i>Misure</i>			
Misura	Descrizione	Aggregabilità	Derivata
ImportoNetto	Quantità * PrezzoUnitario	Additiva	SI
Quantità		Additiva	NO
GiorniPagamento	Giorni trascorsi dall'arrivo della fattura al cliente e il pagamento (DataEmissioneFattura - DataPagamentoFattura)	Semiadditiva	SI
ImportoLordo	ImportoNetto + Tasse	Additiva	SI

### 4.3.2 Progettazione concettuale del *data mart*

Lo schema concettuale del *data mart* fatture è presentato in Figura 4-3, nel quale è possibile notare la presenza di:

- A. *un fatto*: le fatture;
- B. *quattro misure*: ImportoNetto, Quantità, GiorniPagamento, ImportoLordo;
- C. *quattro gerarchie dimensionali*: Cliente, Data, Succursale, Prodotto;
- D. *una dimensione degenera*: NumeroFattura che permette di effettuare analisi con diverso livello di dettaglio, facendo in modo di poter analizzare il fatto fatture, oltre che per righe, anche per le singole fatture.

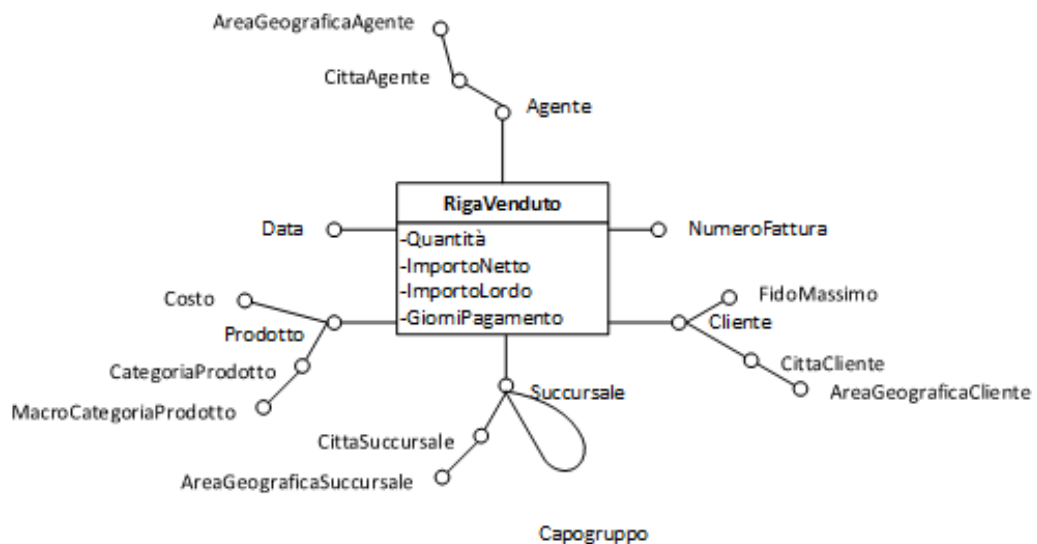


Figura 4-3 – Schema concettuale iniziale del *data mart* Fatture

## 4.4 Il processo delle campagne promozionali

Si descrive la specifica dei requisiti e la progettazione concettuale iniziale del *data mart* campagne promozionali.

Una campagna promozionale rappresenta una promozione, legata al prodotto, che l'azienda attua verso un azienda cliente. Il fatto campagne promozionali è una fattura di cui almeno un prodotto sia collegato a una campagna promozionale. Una fattura può contenere più prodotti legati alle rispettive campagne promozionali.



#### 4.4.1 Specifica dei requisiti del fatto

Nella fase di Specifica dei requisiti, sono stati prodotti dei documenti formali dei requisiti di analisi dei dati richiesti che ne evidenziano le caratteristiche salienti da modellare nella successiva fase della progettazione concettuale.

Nella modulistica prodotta vengono evidenziati i fatti, le misure e le dimensioni coinvolte. Dai requisiti di analisi raccolti in precedenza (Paragrafo 3.4.1) discende la seguente specifica dei requisiti del fatto campagne promozionali che associa a ogni requisito di analisi le dimensioni, le misure e le aggregazioni coinvolte.

*Specifica dei requisiti del fatto Venduto*

N.	Requisito di analisi	Dimensioni	Misure
10	Ricavato da campagne promozionali per sede "Capogruppo".	Sede "Succursale", CampagnaPromozionale NumeroFattura	importoNetto
11	Nuovi clienti derivati da campagne per sede "Capogruppo".	Sede "Capogruppo", Cliente, NumeroFattura	ISNuovoCliente
12	Confronto andamento campagna rispetto al budget preventivato.	CampagnaPromozionale, NumeroFattura	importoNetto
13	Analisi incremento vendite per prodotto e per aziende clienti esistenti.	Prodotto, Cliente, NumeroFattura	Quantità, importoNetto

Si descrive il fatto Campagne promozionali, specificandone la granularità, e quindi il suo significato, le dimensioni e le misure preliminari interessate.

La granularità del singolo fatto determina la dimensione del *data mart* e il tipo di analisi che si può effettuare sui dati; partendo da un dato ad elevata granularità è possibile ottenere altri dati meno dettagliati.

*Dimensioni preliminari del fatto Campagna promozionali*

Descrizione	Dimensioni preliminari	Misure preliminari
La riga della fattura. Ad ogni riga corrisponde un prodotto venduto ad un cliente sotto campagna promozionale	Succursale, Prodotto, Cliente, Data, CampagnaPromozionale, NumeroFattura	Quantità, importoNetto, ISNuovoCliente

Si descrivono le dimensioni del fatto Campagna promozionale specificando per ognuna di esse il nome, una descrizione e la granularità.

*Dimensioni - Riepilogo*

<b>Nome</b>	<b>Descrizione</b>	<b>Granularità</b>
Cliente	L'azienda cliente che a cui è intestata la fattura	Un cliente
Data	Dimensione temporale in cui si svolgono i fatti	Un giorno
CampagnaPromozionale	La campagna promozionale a cui è soggetto il prodotto	Una campagna promozionale
Succursale	La sede "Succursale" a cui fa capo l'agente	Una sede
Prodotto	Il prodotto finito che viene spedito a seguito di un ordine	Un prodotto

Per ogni dimensione vengono elencati gli attributi e una loro descrizione.

*Dimensione - Cliente*

<b>Attributo</b>	<b>Descrizione</b>
RagioneSocialeCliente	Ragione sociale del cliente
CittaCliente	Città del cliente
AreaGeograficaCliente	Area geografica del cliente

*Dimensione - Data*

<b>Attributo</b>	<b>Descrizione</b>
Anno	Anno
Semestre	Semestre
Trimestre	Trimestre
Mese	Mese

*Dimensione - Succursale*

<b>Attributo</b>	<b>Descrizione</b>
RagioneSocialeSuccursale	Ragione sociale della sede
CapoGruppo	Ragione sociale della sede "Capogruppo"
CittaSuccursale	Città della succursale
AreaGeograficaSuccursale	Area geografica della succursale

*Dimensione - Prodotto*

<b>Attributo</b>	<b>Descrizione</b>
SiglaProdotto	Sigla del prodotto
MacroCategoriaProdotto	Nome della macro categoria del prodotto
CategoriaProdotto	Nome della categoria del prodotto

*Dimensione - CampagnaPromozionale*

<b>Attributo</b>	<b>Descrizione</b>
SiglaCampagnaPromozionale	Sigla della campagna promozionale

CostoCampagnaPromozionale	Costo della campagna promozionale
RicavoPrevistoCampagnaPromozionale	Ricavo previsto della campagna promozionale

*Dimensione - NumeroFattura*

Attributo	Descrizione
NumeroFattura	Numero della fattura

Si descrivono le gerarchie dimensionali specificando per ogni dimensione le possibili gerarchie fra gli attributi e il loro tipo (bilanciata, incompleta, ricorsiva).

*Gerarchie dimensionali*

Dimensione	Descrizione gerarchia	Tipo gerarchia
Data	Data-> Mese-> Trimestre-> Semestre-> Anno	Bilanciata
Cliente	Cliente -> CittaCliente -> AreaGeograficaCliente	Bilanciata
Succursale	RagioneSocialeSuccursale-> CittaSuccursale -> AreaGeograficaSuccursale	Bilanciata
Prodotto	SiglaProdotto-> CategoriaProdotto -> MacroCategoriaProdotto	Bilanciata

Di seguito viene descritta la modalità di trattamento delle dimensioni coinvolte nel fatto Campagne promozionali.

*Dimensioni*

Nome	Granularità	Parzialità	Trattamento modifiche
Cliente	Un cliente	NO	Tipo 2
Data	Una data	NO	Tipo 1
CampagnaPromozionale	Una campagna promozionale	NO	Tipo 1
Succursale	Una sede "Succursale"	NO	Tipo 2
Prodotto	Un prodotto	NO	Tipo 2
NumeroFattura	Una fattura	NO	Tipo 1

Si descrive ogni misura del fatto Ordini di vendita, come derivarla da altre misure, se calcolata, e il tipo di aggregabilità.

*Misure*

Misura	Descrizione	Aggregabilità	Derivata
ImportoNetto	Quantità * PrezzoUnitario	Additiva	SI
Quantità		Additiva	NO
ISNuovoCliente	Indica che la fattura, è la prima per quell'azienda cliente	Additiva	SI

ImportoLordo	ImportoNetto + Tasse	Additiva	Si
--------------	----------------------	----------	----

#### 4.4.2 Progettazione concettuale del *data mart*

Lo schema concettuale del *data mart* Campagna Promozionale è presentato in Figura 4-4, nel quale è possibile notare la presenza di:

- A. *un fatto*: le Campagne Promozionali;
- B. *tre misure*: ImportoNetto, Quantità, ISNuovoCliente;
- C. *cinque gerarchie dimensionali*: Cliente, Data, Succursale, Prodotto, Campagna Promozionale;
- D. *una dimensione degenera*: NumeroFattura.

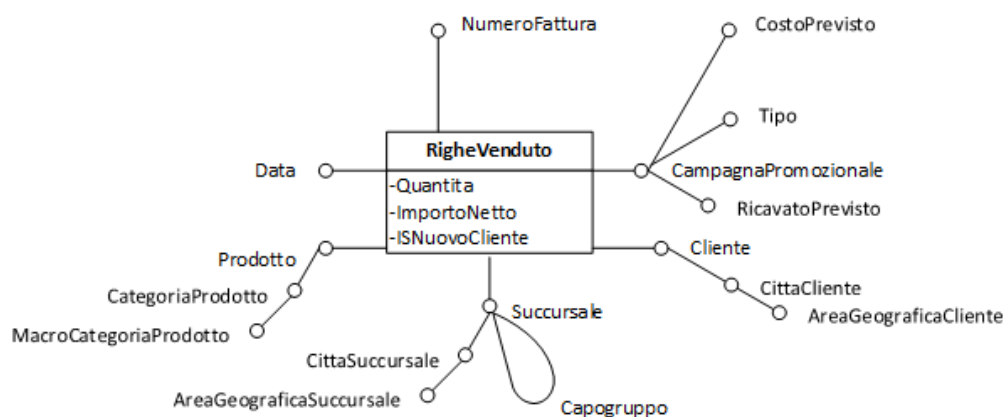


Figura 4-4 - Schema concettuale iniziale del *data mart* Campagne promozionali

#### 4.5 Tabelle riepilogative delle dimensioni e delle misure

Prima di procedere con la progettazione concettuale a partire dai dati operazionali, la redazione dei *data mart* finali e la conclusiva progettazione logica del *data warehouse*, si presenta una tabella riepilogativa che illustra quali sono le dimensioni e le misure condivise da più fatti.

Riepilogo - Dimensioni dei fatti			
Dimensione	Fatturato	Ordini	CampagnePromozionali
Agente	X	X	

Cliente	X	X	X
Data	X	X	X
Succursale	X	X	X
CampagnaPromozionale			X
Prodotto	X	X	X
NumeroOrdine		X	
NumeroFattura	X		X

*Riepilogo - Misure dei fatti*

<b>Misura</b>	<b>Fatturato</b>	<b>Ordini</b>	<b>CampagnePromozionali</b>
ImportoNetto	X	X	X
GiorniPagamento	X		
PrezzoUnitario		X	
Quantità	X	X	X
Tasse		X	
ImportoLordo	X	X	X
ISNuovoCliente			X

## **5 PROGETTAZIONE CONCETTUALE FINALE E LOGICA DEI DATA MART E DEL DATA WAREHOUSE**

Dopo aver presentato il *database* operativo (Paragrafo 2.3) e gli schemi concettuali iniziali dei *data mart* (Paragrafo 4.2.2, Paragrafo 4.3.2, Paragrafo 4.4.2,), in questo capitolo verranno definite le entità (evento, componente e classificazione) e gli schemi concettuali finali dei *data mart*. Infine verranno presentati gli schemi logici dei singoli *data mart* e del *data warehouse*.

## 5.1 Analisi dei dati operazionali

Sulla base dei requisiti di analisi dei dati (Paragrafo 3.2.1, Paragrafo 3.3.1, Paragrafo 3.4.1), si esamina lo schema relazionale per decidere quali tabelle e attributi sono interessanti per le analisi. In questa prima fase non sono state considerate le chiavi primarie ed esterne in quanto verranno discusse più avanti.

Tutte le tabelle e gli attributi non rappresentati in Figura 2.2 devono considerarsi scartati, mentre per tutti quelli esistenti valgono le considerazioni:

- **C\_CAMPAIGN:** si mantiene *startdate*, *enddate*, *name*, *description*, *costs*, *c\_channel*. Anche se non è richiesto esplicitamente è bene conservare anche l'informazione *c\_channel* in quanto contiene il canale della campagna promozionale (il motivo verrà spiegato più avanti).
- **C\_INVOICE:** si mantiene *dateInvoiced*. Questa tabella viene unita, tramite il processo di ETL, con la tabella C\_INVOICELINE in modo da riportare *dateInvoiced* su tutte le righe della fattura.
- **C\_INVOICELINE:** si mantiene *priceactual*, *qtyinvoiced*, *linenetamt* e *taxamt*.
- **C\_PAYMENTTERM:** si mantiene *netdays*, per calcolare in fase di ETL la data di scadenza del pagamento.
- **FIN\_PAYMENT:** si mantiene *paymentdate*, *amount*, per calcolare in fase di ETL l'ammontare pagato, riferito ad una fattura e la data dell'ultimo pagamento effettuato dall'azienda cliente.
- **C\_BPARTNER:** si mantiene *ISsalesrep*, *ISemployee*, *IScustomer* da utilizzare in fase di ETL per poter estrapolare i dati che riguardano, rispettivamente, i rappresentanti, gli impiegati e i clienti. Inoltre si mantiene *Name*, *so\_creditused* e *so\_creditlimit*.
- **C\_BPARTNER\_LOCATION:** non contiene informazioni utili in quanto è un contenitore di chiavi esterne che si riferiscono a C\_LOCATION.
- **C\_LOCATION:** si mantiene *city*.
- **C\_REGION:** si mantiene *name*.

- **AD\_CLIENT:** si mantiene *name*.
- **AD\_ORG:** si mantiene *name*.
- **AD\_ORG\_INFO:** non contiene informazioni utili in quanto è un contenitore di chiavi esterne che si riferiscono a *C\_BPARTNER\_LOCATION*.
- **M\_PRODUCT:** si mantiene *ISold* che consente in fase di ETL di separare la tabella in prodotti non venduti e prodotti venduti. Inoltre si mantiene *Value, Name, Cost, Unitperpallet*.
- **M\_PRODUCT\_CATEGORY:** si mantiene *Name* e *producttype* che consentono in fase di ETL di creare una gerarchia sui prodotti.
- **C\_ORDER:** si mantiene *DocumentNo*, che rappresenta il numero dell'ordine di vendita. Questa tabella viene unita, tramite il processo di ETL, con la tabella *C\_ORDERLINE* in modo da riportare il dato su tutte le righe della fattura.
- **C\_ORDERLINE:** si mantiene *dateordered, datepromised, datedelivered, priceactual, qtyordered* e *taxamt*.

A questo punto vengono classificate le tabelle interessanti in base al loro contenuto ed alle loro relazioni, come entità evento, entità componente e entità di classificazione.

- **Entità evento.** Sono le tabelle che descrivono eventi che si verificano frequentemente e che contengono attributi numerici che potrebbero essere delle misure. In questa categoria rientra l'accorpamento di *C\_ORDERLINE* e *C\_ORDER* con la granularità *C\_ORDERLINE* e l'accorpamento di *C\_INVOICELINE* e *C\_INVOICE* con la granularità *C\_INVOICELINE*.
- **Entità componente.** Sono le tabelle in relazione 1:N con le entità evento e ne definiscono i dettagli; analizzando le entità evento selezionate si scoprono le seguenti entità componente:
  - Per l'entità *C\_ORDERLINE*, accorpata con *C\_ORDER*, le entità componenti sono Prodotto (derivata dal partizionamento di *M\_PRODUCT*), Cliente



(derivata dal partizionamento di *C\_BPARTNER*), Succursale (derivata da *AD\_CLIENT*), Agente (derivata dal partizionamento di *C\_BPARTNER*).

- Per l'entità *C\_INVOICELINE*, accorpata con *C\_INVOICE*, le entità componenti sono Prodotto (derivata dal partizionamento di *M\_PRODUCT*), Cliente (derivata dal partizionamento di *C\_BPARTNER*), Succursale (derivata da *AD\_CLIENT*), Agente (derivata dal partizionamento di *C\_BPARTNER*).
- Per l'entità Campagne promozionali, le entità componenti sono Prodotto (derivata dal partizionamento di *M\_PRODUCT*), Cliente (derivata dal partizionamento di *C\_BPARTNER*), Succursale (derivata da *AD\_CLIENT*), CampagnaPromozionale (derivata da *C\_CAMPAIGN*).

Va aggiunto che tra le entità componente di ogni entità si aggiunge quella che rappresenta il tempo (presente nello schema relazionale con attributi di tipo *Date*).

- **Entità di classificazione.** Sono le tabelle in relazione 1:N con le entità componente e scaturiscono da gerarchie nei dati.
  - Per l'entità Cliente, le entità di classificazione interessanti sono *C\_LOCATION* e *C\_REGION*.
  - Per l'entità Prodotto, le entità di classificazione interessanti sono *M\_PRODUCT\_CATEGORY*.
  - Per l'entità Succursale le entità di classificazione interessanti sono *AD\_ORG* e *AD\_CLIENT*.

## 5.2 Integrazione dei dati operazionali negli schemi concettuali dei *data mart* iniziali

Il confronto degli schemi concettuali dei *data mart* iniziali con i dati operazionali ha portato alla definizione degli schemi concettuali finali.

- Alla dimensione Data, condivisa fra tutti i *data mart*, è stata aggiunta la gerarchia, Mese, Trimestre, Semestre, Anno.
- Alla dimensione Prodotto, condivisa fra tutti i *data mart*, sono stati aggiunti i campi: Sigla, Descrizione, UnitàImballo.
- Alla dimensione Cliente e Succursale, condivise fra tutti i *data mart*, è stato aggiunto il campo RagioneSociale.
- Alla dimensione CampagnaPromozionale, sono stati aggiunti i campi: nome, canale, dataInizioCampagna, dataFineCampagna.
- Sono stati fusi i *data mart* “Venduto” e “Campagne Promozionali” in quanto hanno la tabella dei fatti che varia solo di poche misure.
- Ad una prima analisi i *data mart* “Ordini di vendita” e “Venduto”, risultano simili, ma dopo una discussione con il committente si è scelto di lasciarli separati perché:
  - Non tutti gli ordini vadano a buon fine. Quindi bisogna poter analizzare la differenza tra il venduto e l’ordinato.
  - Ad un ordine possono corrispondere una o più fatture, a cui corrisponde un comportamento diverso da parte del cliente.

## 5.3 Ordini di Vendita

Dopo aver analizzato la sorgente dati, si definisce la dimensionalità potenziale del fatto Ordini di vendita che ammonta a circa 17.000 di record.

### 5.3.1 Progettazione concettuale finale del *data mart*

In Figura 5-1 viene mostrato lo schema concettuale finale del *data mart* Ordini di vendita.

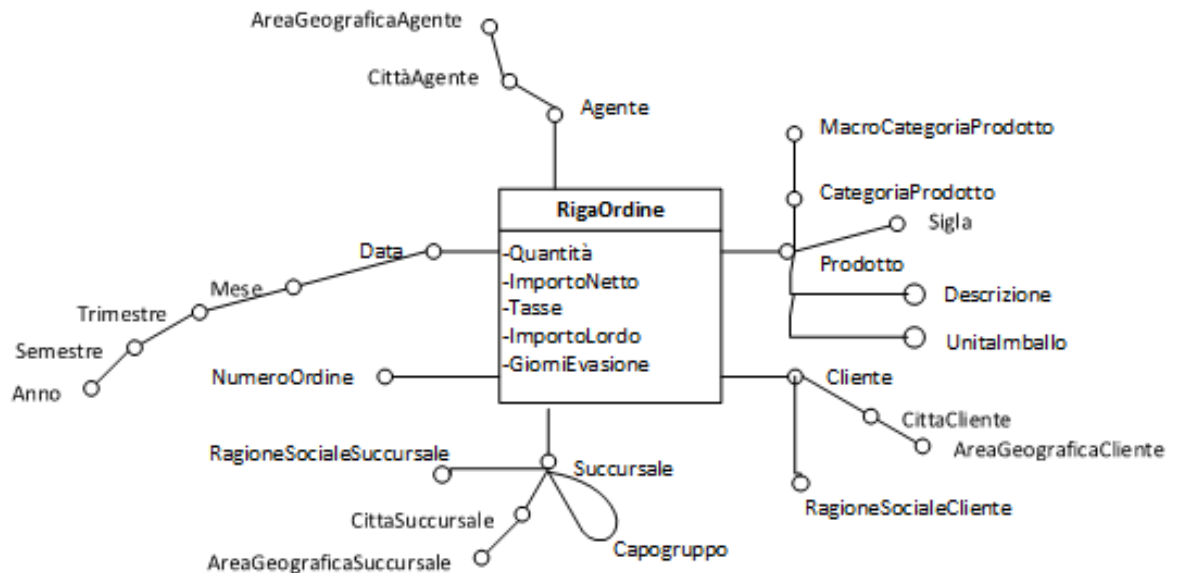


Figura 5-1 - Schema concettuale finale del data mart Ordini di vendita

### 5.3.2 Progettazione logica del *data mart*

Prima di mostrare lo schema logico del data mart Ordini di acquisto è bene ricordare i passi salienti della progettazione logica dei *data mart* e del *data warehouse*.

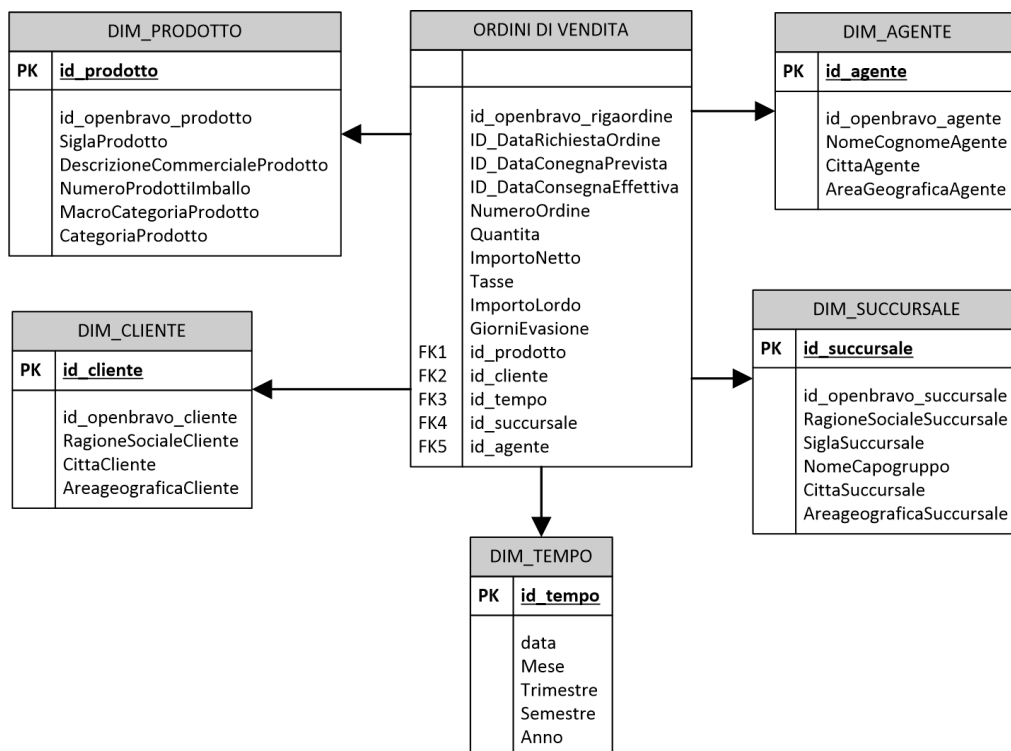
Come proposto da [Alb12], in questa fase si passa dagli schemi finali dei *data mart* allo schema relazionale del *data warehouse*, trasformando prima i tre schemi concettuali finali dei *data mart* nei tre schemi relazionali a stella, e poi integrando i vari schemi dei *data mart* in un unico schema del data warehouse.

Per le tre tabelle dei fatti viene creata una tabella relazionale, con una chiave primaria surrogata e tutte le misure come attributi.

Successivamente viene creata una tabella relazionale per ogni dimensione presente nello schema concettuale finale. Queste tabelle contengono una chiave primaria surrogata e tutti gli attributi presenti nella rispettiva dimensione.

Viene aggiunta quindi alle tabelle dei fatti una chiave esterna per ogni dimensione.

In Figura 5-2 viene mostrato lo schema logico del *data mart* Ordini di vendita.



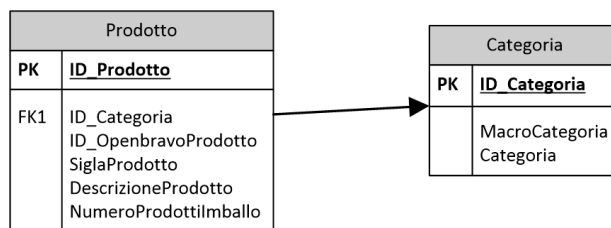
**Figura 5-2 - Schema logico del *data mart* Ordini di vendita**

Il *database* rappresentato in Figura 5-2 è detto schema a stella in quanto esiste una tabella dei fatti in relazione con più tabelle dimensionali. Questo schema verrà utilizzato anche dai restanti schemi.

Si è preferito uno schema a stella a discapito di quello a fiocco di neve (dove vi è una tabella dei fatti in relazione con più tabelle dimensionali, alcune delle quali sono in relazione con altre tabelle dimensionali) in quanto [Kim02]:

- L'inserimento di molte tabelle con schema a fiocco di neve rende più complessa la presentazione, portando l'utente a relazionarsi con uno strumento più complesso da comprendere.
- L'utilizzo di un *database* che adotta uno schema a fiocco di neve rende più difficoltosa la sua ottimizzazione e l'utilizzo di numerose tabelle e *joins* rallenta l'esecuzione delle query, provocando una diminuzione delle performance.

- Lo spazio risparmiato con l'utilizzo di uno schema a fiocco di neve è irrilevante, infatti, prendendo l'esempio della tabella DIM\_Prodotto, se, per assurdo, si fosse adottato lo schema di Figura 5-3, avremo il campo ID\_Categoria (che occupa



**Figura 5-3 - DIM\_Prodotto con schema a fiocco di neve**

solo 2 byte) al posto dei campi CategoriaProdotto e MacroCategoriaProdotto (che occupano complessivamente 40 byte), con un risparmio di 36 byte per riga. La tabella DIM\_Prodotto ha circa 1.500 righe, quindi con un rapido calcolo ( $1.500 * 36$  byte) avremo un risparmio totale di 0.54 MB (supponendo che la quantità di memoria che occupa la tabella Categoria sia irrisoria), che è una cifra insignificante in confronto con la tabella dei fatti che è grande circa 1 GB.

L'utilizzo dello schema a stella porta:

- Nella tabella DIM\_Prodotto: l'inserimento degli attributi descrittivi "MacroCategoria" e "Categoria", che permetterà, in seguito di poter creare una gerarchia.
- Nelle tabelle DIM\_Cliente, DIM\_Agente, DIM\_Succursale: l'inserimento degli attributi descrittivi "AreaGeografica" e "Citta", che permetterà, in seguito di poter creare una gerarchia.
- Nella tabella DIM\_Succursale: si è inserito l'attributo descrittivo "NomeCapogruppo", che permetterà, in seguito di poter creare una gerarchia.

Come si evince dalla Figura 5-2, la dimensione degenera NumeroOrdine diventa attributo della tabella del fatto Ordini di Vendita.

A causa del trattamento delle modifiche di tipo 2 nelle dimensioni Cliente, Succursale, Agente è necessario fare uso di una chiave primaria surrogata.

Questa soluzione è ottimale per effettuare corrette analisi multidimensionali perché risulta possibile stabilire che record con chiavi surrogate diverse riguardano la stessa entità che può aver cambiato residenza.

## 5.4 Venduto e Campagne promozionali

A partire da questo paragrafo il *data mart* nato dalla fusione del *data mart* “Venduto” e quello “Campagne Promozionali” verrà chiamato “Venduto” per una maggiore chiarezza.

Dopo aver analizzato la sergenti dati, si definisce la dimensionalità potenziale del fatto Venduto che ammonta a circa 15.000 di record, di cui 2.000 sono condivisi con il fatto “Campagne Promozionali”.

### 5.4.1 Progettazione concettuale finale del *data mart*

In Figura 5-4 viene mostrato lo schema concettuale finale del *data mart* Venduto.

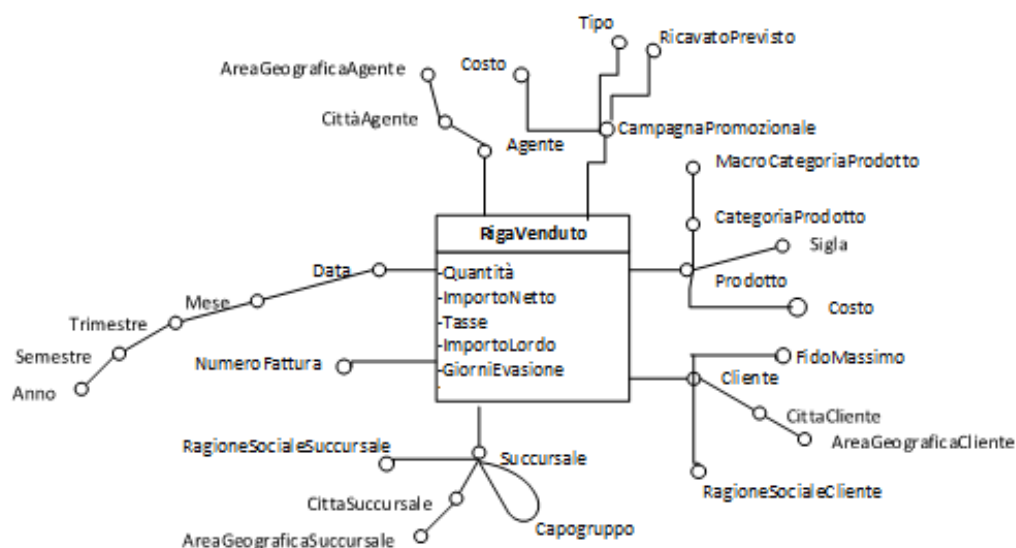
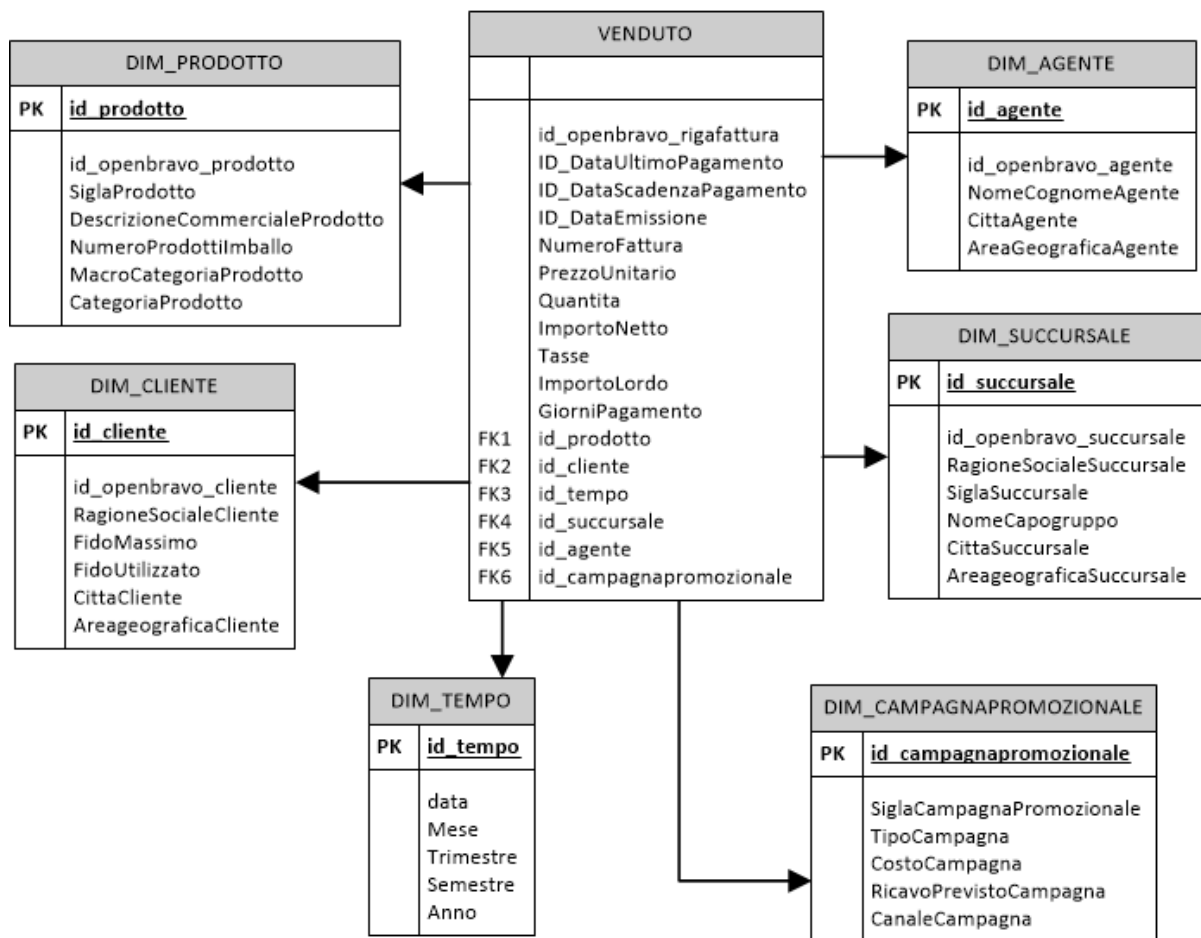


Figura 5-4 - Schema concettuale finale del *data mart* Venduto

### 5.4.2 Progettazione logica del *data mart*

In Figura 5-5 viene mostrato lo schema logico del *data mart* Venduto.



**Figura 5-5 - Schema logico del data mart Venduto**

Come si evince dalla Figura 5-5 , la dimensione degenera NumeroFattura diventa attributo della tabella del fatto Venduto.

Per le dimensioni Cliente, Succursale, Agente valgono le considerazioni riguardanti il trattamento delle modifiche di tipo 2 (paragrafo 5.3.2).

## 5.5 Progettazione logica del *data warehouse*

Osservando gli schemi logici dei due *data mart* si può vedere che hanno in comune diverse dimensioni.

Solitamente fondere le dimensioni in comune in una sola tabella può comportare la creazione di viste per rinominare o escludere alcuni attributi: in questo caso non si hanno problemi del genere e la fusione può essere fatta direttamente.

In Figura 5-6 si presenta lo schema logico complessivo del *data warehouse* con le dimensioni in comune e condivise dai due *data mart*.

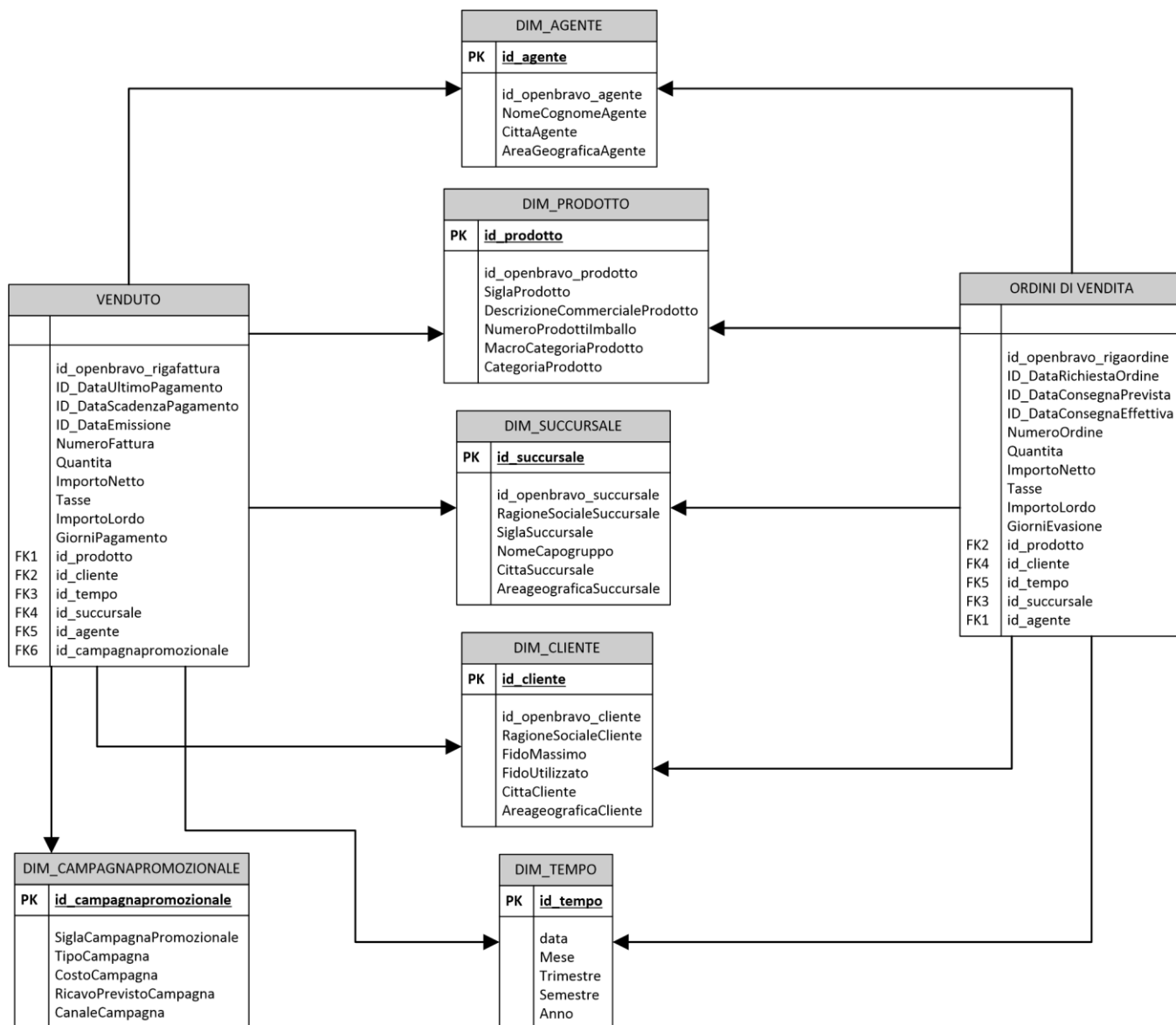


Figura 5-6 – Schema logico complessivo del data warehouse



## 6 STRUMENTI DI SVILUPPO

In questo capitolo verranno elencanti gli strumenti utilizzati nel processo di realizzazione del sistema di *Business Intelligence*.

Nella prima parte verrà fatta una breve panoramica sui termini utilizzati.

Successivamente verrà fornita una visione specifica della piattaforma Pentaho, che racchiude gli strumenti di ETL, analisi multidimensionale e reportistica ed infine gli strumenti utilizzati per l'amministrazione dei *database*.

## 6.1 Glossario

In questo paragrafo sono presenti i termini utilizzati nel corso del capitolo per rendere più agevole la lettura.

### 6.1.1 OLAP

Il termine OLAP (acronimo di *On-Line Analytical Processing*) si usa per riferirsi all'attività di analisi multidimensionali di grandi quantità di dati con modalità interattive e intuitive per cambiare le prospettive di analisi e per passare a livelli diversi di sintesi dei dati dettagliati [Alb12].

L'OLAP *client* interagisce con il gestore dei dati con una delle seguenti principali modalità (Figura 6-1) [Alb12]:

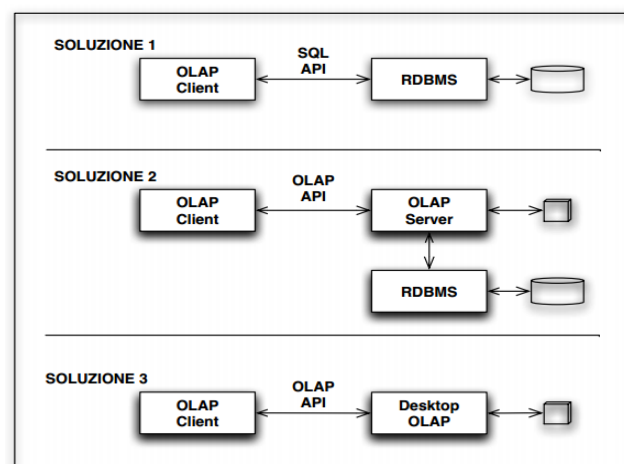


Figura 6-1 - Soluzioni per sistemi OLAP [Alb12]

**Soluzione 1.** Il *data warehouse* è memorizzato in un sistema relazionale (*Data server*) e le interazioni con l'*OLAP client* avvengono in SQL. Il beneficio di questa soluzione è che si usa una tecnologia standard di solito già disponibile. Nel passato l'approccio non era ritenuto soddisfacente per le prestazioni dei RDBMS come sistemi per *data warehouse* e i limiti dell'SQL come linguaggio OLAP. Ma ormai i principali produttori di RDBMS hanno reso i sistemi sempre più specializzati per applicazioni OLAP (*OLAP-Aware RDBMS*), rendendoli consapevoli di gestire *data*

*warehouse* con schemi relazionali particolari (a stella, a fiocco di neve o a costellazione), con gerarchie dimensionali, ed estendendoli con strutture di memorizzazione specifiche e con la gestione di viste materializzate (dette anche *aggregati*, *MQT (materialized query tables)*, *summary data*).

**Soluzione 2.** L'*OLAP client* interagisce con un *OLAP server*, un sistema che fornisce una visione multidimensionale a cubo dei dati di un *data mart* analizzabili con le tipiche operazioni *slice*, *dice*, *drill down*, *roll up*, *pivot*. Un *OLAP server* può essere di uno dei seguenti tipi:

1. *MOLAP*, il quale mantiene nella memoria locale sia i dati del cubo, prelevati dal *Data server*, che gli aggregati del cubo esteso (viste materializzate), utilizzando opportune strutture a matrici. La soluzione fornisce ottime prestazioni, ma non è adatta per grandi quantità di dati.
2. *ROLAP*, che memorizza sia i dati del cubo che gli aggregati del cubo esteso nel *Data server*.
3. *HOLAP (Hybrid OLAP)*, che memorizza nella memoria locale gli aggregati del cubo esteso in opportune strutture a matrici e lascia i dati del cubo nel *Data server* (utilizzato dalla piattaforma Pentaho).

**Soluzione 3.** L'*OLAP client* interagisce con un sistema locale *DOLAP (Desktop OLAP)* che gestisce piccole quantità di dati estratti dal *OLAP server*, dal *Data server* o da un DBMS operativo. Il principale prodotto di questo tipo è SAP BusinessObjects™. Il sistema non mantiene aggregati nella memoria locale, ma solo i risultati delle ultime operazioni. Gli aggregati vengono calcolati dall'*OLAP server*, dal *Data server* o da un DBMS operativo.

### 6.1.2 MDX

MDX è l'acronimo di *Multi-Dimensional eXpressions* (progettato da Microsoft e introdotto in SQL Server Analysis Service 7.0 nel 1998). Nonostante sia un linguaggio ancora non standardizzato, è adottato dalla maggior parte dei venditori OLAP ed è diventato lo standard di fatto per tali sistemi.

MDX è utilizzato per l'interrogazione e la manipolazione dei dati memorizzati in cubi OLAP multidimensionali. E' un linguaggio che non permette la creazione di strutture dati (come lo SQL) ma solo la loro interrogazione.

Una interrogazione MDX consta una richiesta di dati (*SELECT*), un'origine (*FROM*) e un filtro (*WHERE*). Queste parole chiave rappresentano, insieme ad altre, gli strumenti utilizzati per estrarre parti specifiche di dati da un cubo per effettuare analisi.

La visualizzazione di un set di risultati di MDX può includere più di tre dimensioni e visualizzarne la struttura può essere estremamente difficile; per tale motivo MDX utilizza una sintassi estremamente specifica e uniforme per fare riferimento alle celle di dati, indipendentemente dal fatto che i dati formino una singola cella o un gruppo di celle.

MDX supporta due modi distinti per interrogare i *database* multidimensionali:

1. *usato in un'espressione*, definisce e manipola oggetti multidimensionali e dati per calcolare i valori;
2. *usato come un linguaggio d'interrogazione*, recupera i dati dai *database* OLAP.

L'esempio seguente, permette di vedere una semplice interrogazione che ha come risultato il "Totale delle vendite su internet" in Australia, nel secondo trimestre del 2003, per la categoria di prodotto *Mountain*:<sup>3</sup>

```
SELECT Measures.[Internet Sales Amount] ON COLUMNS
FROM [Adventure Works]
WHERE ( [Date].[Calendar].[Calendar Quarter]. & [2003] & [2],
[Product].[Product Line].[Mountain],
[Customer].[Country].[Australia] )
```

Come le tabelle e le colonne sono la base del linguaggio SQL, dimensioni, gerarchie e livelli sono il fulcro di MDX. Come si evince dall'esempio, dopo aver scelto i dati da visualizzare (*WHERE*), questo linguaggio permette di visualizzare l'informazione (aggregata o non) per riga (*ON ROWS*) e/o per colonna (*ON COLUMNS*).

---

<sup>3</sup> Esempio tratto da "Introducion to MDX"

### 6.1.3 PostgreSQL

PostgreSQL è un ORDBMS (*Object Relational Database Management System*) e possiede un'architettura *client/server*.

La differenza tra un RDBMS e un ORDBMS, è che il secondo oltre ad essere un *Relational Database Management System (RDBMS)*, supporta nativamente negli schemi relazionali e nelle query strutture e modelli *object-oriented* come oggetti, classi, ereditarietà.

E' *Open source* e conforme agli standard ansiSQL (SQL 2003) supportandone caratteristiche sintattiche e semantiche.

Una sessione PostgreSQL è caratterizzata dalla cooperazione dei seguenti processi [Pos12]:

1. *Un processo server*, chiamato *postgres*, che gestisce i *database*, accetta le connessioni al database da parte dell'applicazione *client* ed esegue le operazioni sui *database* in base alle richieste dei client.
2. *Un applicazione client*, che interfaccia i client con il *database* permettendo di effettuare le operazioni di lettura e mantenimento dei *database*.

Essendo un'applicazione *client/server*, questi dovranno comunicare tra loro tramite una connessione di rete.

Il *server* può accettare connessioni concorrenti multiple dai *client* avviando un processo diverso per ogni connessione (se la connessione cade il processo termina), che permette ai *client* di non avere interferenze o rallentamenti nella comunicazione con i *database*.

Questa soluzione permette l'isolamento dei processi e delle risorse allocate per ciascuna sessione, garantendo una migliore sicurezza ed affidabilità. Allo stesso tempo si può andare incontro ad uno spreco, se queste risorse non sono completamente sfruttate dall'utente.

## 6.2 Pentaho

Pentaho è una piattaforma di *Business Intelligence (BI)* creata nel 2004 da un team di professionisti provenienti da aziende leader del settore, con l'obiettivo di offrire un'alternativa *Open source* alle piattaforme proprietarie già esistenti, proponendo diverse caratteristiche e benefici rispetto alle offerte commerciali.

A tale scopo sono stati integrati in un'unica piattaforma i popolari progetti *Open source* come JFreeReport, Kettle, Mondrian e Weka.

Come la maggior parte dei progetti *Open source*, anche Pentaho ha la sua *community* di sviluppatori.

A differenza di altri progetti *Open source*, nel caso di Pentaho, si trova un'azienda madre (*corporate*) che coordina e governa lo sviluppo dei componenti della *suite* dettando le evoluzioni e i tempi di sviluppo.

La suite Pentaho esiste in due versioni: quella *Community* (usata in questo progetto) e quella *Professional*.

La differenza sostanziale sta nel fatto che Pentaho offre nella versione *Professional*, sotto versamento di una quota annua da parte del cliente, strumenti aggiuntivi e supporto tecnico, non presenti nella versione *Community*.

### 6.2.1 La suite Pentaho Community Edition

La suite *Pentaho Community Edition* (da adesso in poi Pentaho) mette a disposizione una collezione di strumenti che permettono di sviluppare soluzioni di *Business Intelligence* coprendo tutto il ciclo di vita dei dati.

I moduli che compongono Pentaho (Figura 6-2) possono essere classificati secondo criteri e punti di vista differenti, ma per brevità verranno riportati solo quelli principali [Bou09]:

- **Funzionalità.** Un primo modo per categorizzare i vari componenti è tramite la loro funzionalità. Con questo termine si intende l'attività o le attività per le quali un programma è stato progettato. Dalla prospettiva dell'utente la

funzionalità è ciò che definisce lo scopo del programma stesso e a tale proposito Pentaho offre quelle classiche di Business Intelligence quali *ETL*, *Reporting*, *OLAP*, *data mining* e molte altre funzionalità.

- **Server, Web Client e Desktop.** Vengono etichettati come applicazioni *Server* tutte quelle applicazione eseguite su una macchina centrale accessibile dai *web client* tramite *internet* o *intranet*. Tutte le altre applicazioni che non possono essere definite come *Server* vengono definite come *Desktop*. Sono queste le applicazioni che i progettisti e sviluppatori utilizzano per inviare richieste ai programmi *Server* con il quale sono collegati.

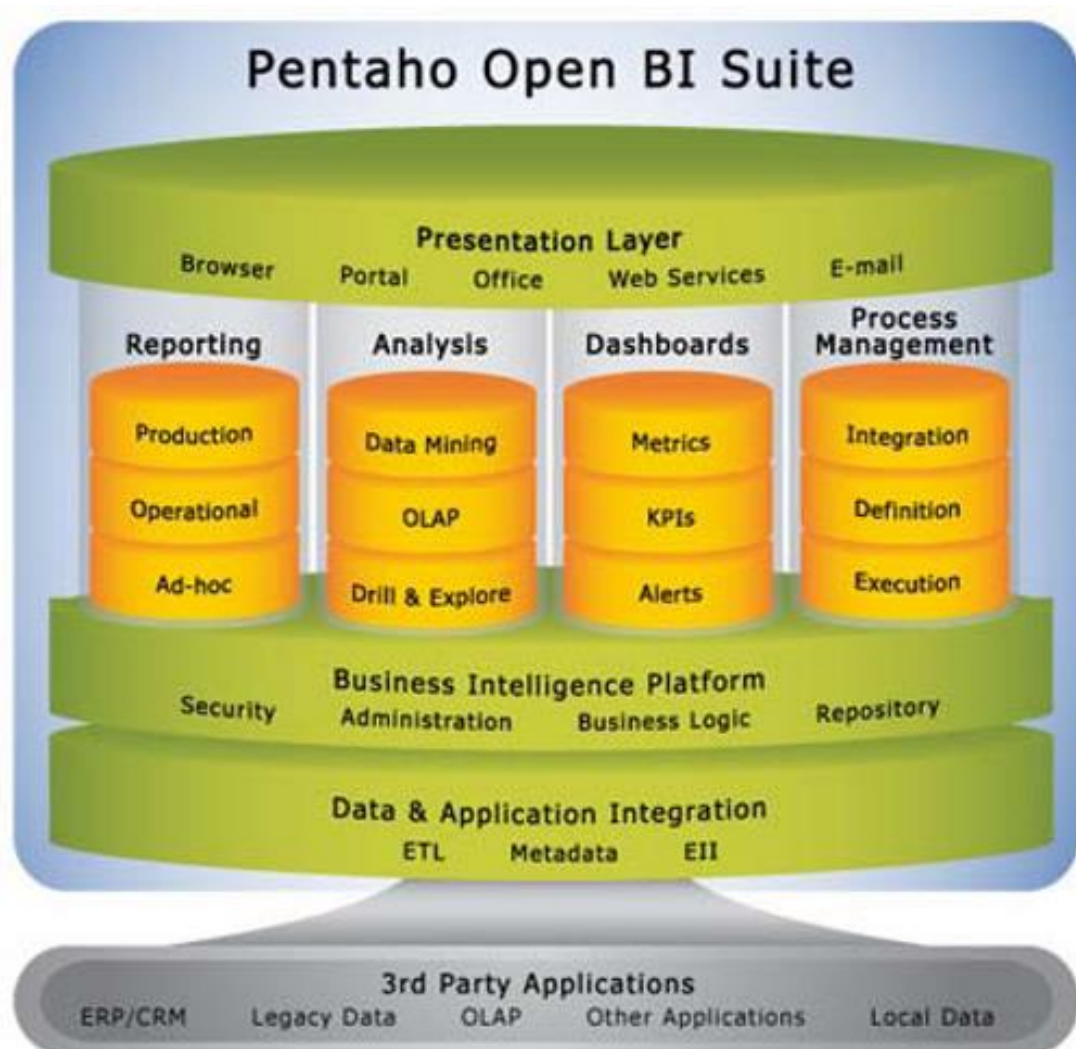


Figura 6-2 - La suite Pentaho

Un tipico esempio di distinzione tra applicazioni *Server*, *client*, *Desktop* è la progettazione, la pubblicazione e l'esecuzione di un documento di *report*. In questo caso *Report Designer* è l'applicazione *Desktop* utilizzata per creare il *report*. Questo sarà pubblicato sul *Pentaho Server*, che potrà essere creato su richiesta dell'utente e visualizzato in output grazie a *Pentaho Web Portal* (il *Client*).

- **Tecnologia di base.** Quasi tutti i programmi contenuti in Pentaho sono stati implementati utilizzando il linguaggio di programmazione Java, mentre alcune componenti lato server implementano applicazioni *web based* basate su tecnologia AJAX. Questa distinzione, a differenza delle altre, è destinata soprattutto agli amministratori di sistema e sviluppatori. Questa puntualizzazione risiede nel fatto che una delle proprietà più rilevanti della piattaforma Java riguarda l'estrema flessibilità e portabilità delle sue applicazioni sulle varie architetture hardware e sistemi operativi. Conseguentemente Pentaho è utilizzabile su diverse piattaforme.

#### **6.2.1.1 Applicazioni lato server**

Il server di Pentaho è un insieme di applicazioni per permettere alla *suite* di Pentaho di svolgere le attività principali. Le applicazioni sono implementate come *Java servlets*, quindi non posso essere eseguite in modo standard, ma hanno bisogno di un contenitore, chiamato *servlet container*, che può essere un HTTP Server (o parte di esso).

Normalmente i *servlet container* vengono eseguiti su un computer in remoto che risponde alle richieste dell'applicazioni *client* connesse al *server* attraverso la rete.

### **Mondrian**

Mondrian è il nome del progetto *Open source* di *Pentaho Analysis Services* ed è scritto in Java.



Mondrian è un motore OLAP ibrido (Hybrid OLAP), quindi utilizza la soluzione 2 descritta nel paragrafo 6.1.1.

In particolare la strategia di aggregazione utilizzata da Mondrian è la seguente:

1. La tabella dei fatti e delle dimensioni vengono memorizzate nel DBMS;
2. Viene usata la clausola “*group by*” di SQL per l’aggregazione dei dati;
3. Se l’ RDMBS supporta le viste materializzate e l’amministratore del *database* sceglie di creare viste materializzate per particolari aggregazioni, allora Mondrian usa, se presente, la vista materializzata più economica.

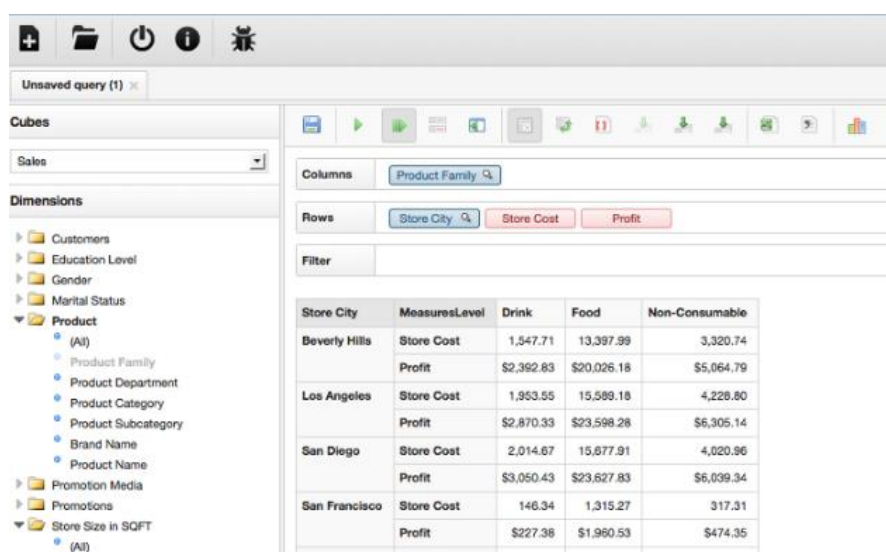
Nella Tabella 6-1 vengono rappresentati i punti di forza e di debolezza dello strumento.

Punti di forza	Punti di debolezza
Si tratta di un motore in ambiente Java ampiamente usato e accreditato.	Difficile da configurare per avere ottime performance.
Garantisce buone performance prestazionali.	

**Tabella 6-1 - Punti di forza e debolezza di Mondrian**

## Saiku

Saiku è un’applicazione *web based* che permette agli utenti di analizzare i dati aziendali e creare e condividere *report* (la sua interfaccia è rappresentata in Figura 6-3).



**Figura 6-3 - Interfaccia di Saiku**

Saiku è un client OLAP in grado di connettersi a diversi server quali Mondrian, Microsoft Analysis Services, SAP BW e Oracle Hyperion.

Grazie all'utilizzo della libreria java OLAP4J, Saiku è in grado di interfacciarsi con cubi OLAP, riuscendo ad effettuare le tipiche operazioni di navigazione degli strumenti OLAP, quali:

- *slice and dice*;
- *drill down e roll up*;
- *ordinamento e filtro dei dati*;
- *pivot*.

Sfruttando le potenzialità di Mondrian, Saiku è in grado di analizzare grandi quantità di dati che, successivamente, possono essere distribuiti attraverso i computer sulla rete locale.

Nella Tabella 6-2 vengono rappresentati i punti di forza e di debolezza dello strumento.

Punti di forza	Punti di debolezza
Interfaccia molto semplice e intuitiva.	Poco scalabile con grossi volumi di dati.
Possibilità di esportare i dati nei formati più comuni come CSV e EXCEL.	Alcune funzionalità migliorabili nell'usabilità da parte dell'utente.
Velocità nei tempi di risposta.	

**Tabella 6-2 - Punti di forza e debolezza di Saiku**

#### **6.2.1.2 Applicazioni desktop**

Le applicazioni desktop della suite Pentaho sono tutte quelle che consentono di interagire direttamente con i dati aziendali senza passare attraverso un server, utilizzando l'applicazione installata in locale sulla macchina dello sviluppatore.

Rientrano in questa categoria i prodotti della suite: Pentaho Data Integration (o Kettle), Pentaho Report Designer, Pentaho Schema Workbench, Pentaho Design Studio, Pentaho Metadata Editor.

## Kettle

Kettle è l'acronimo di *Kettle Extraction Transformation Transportation and Loading Environment*, e nella suite di Pentaho prende il nome di *Pentaho Data Integration* ed è lo strumento che permette di creare processi di ETL.

Kettle è composto da cinque applicazioni differenti:

- **Spoon:** interfaccia grafica che consente di modellare, visivamente, il flusso dei dati dalla sorgente fino alla destinazione, attraverso le trasformazioni e i *job*;
- **Pan:** strumento a linea di comando che consente di interpretare ed eseguire direttamente le trasformazioni disegnate con Spoon;
- **Kitchen:** strumento a linea di comando utilizzato per lanciare e schedare l'esecuzione dei *jobs* disegnati con Spoon (molto usato per la pianificazione delle trasformazioni);
- **Carte:** *web server* che permette l'esecuzione di trasformazioni e/o *jobs* in remoto.

Riassumendo, una procedura ETL in Kettle è un file XML prodotto dal suo editor grafico (Spoon) e, in presenza di una rete, gli utenti possono avviare procedure ETL in remoto usando lo strumento Carte.

Nella Tabella 6-3 vengono rappresentati i punti di forza e di debolezza dello strumento.

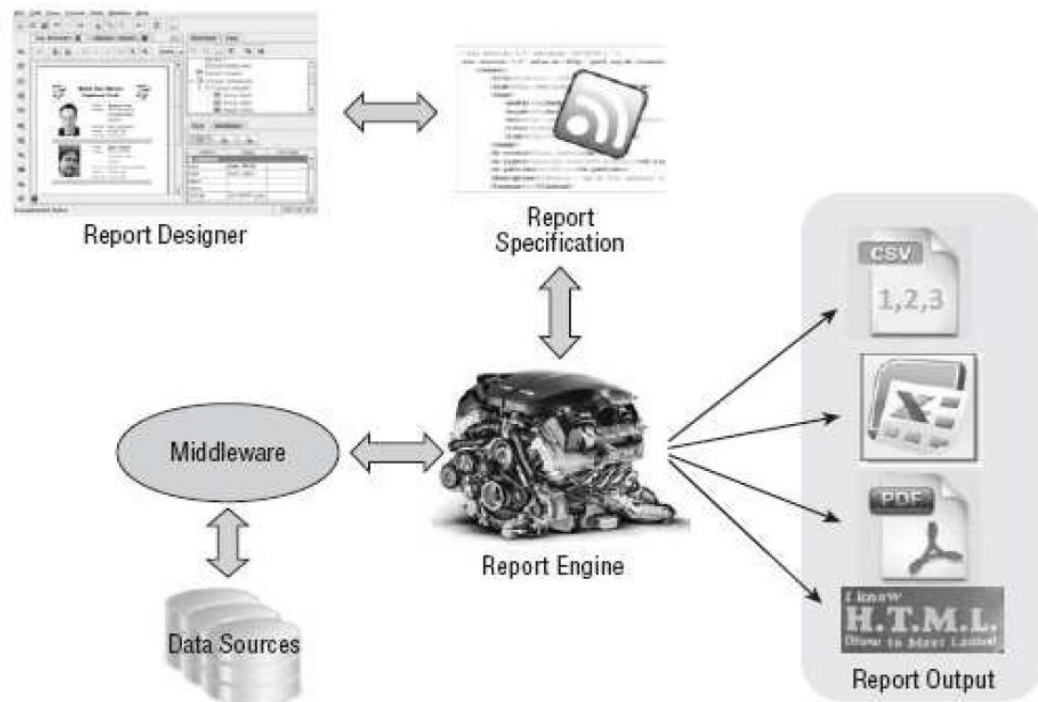
Punti di forza	Punti di debolezza
Facilità di utilizzo.	Nonostante goda di buone performance, sarà sempre più lento rispetto a script SQL mirati.
Facilità di manutenzione dei progetti.	In caso di alto volume dei dati, le performance possono essere inferiori rispetto ad altre soluzioni commerciali.
Flessibilità nello svolgere le trasformazioni.	Per flussi complessi bisogna utilizzare più operazioni per migliorare le prestazioni.
Essendo uno dei primi <i>software ETL open source</i> , ha una vasta comunità di supporto. Può essere utilizzato anche al di fuori della piattaforma Pentaho essendo indipendente dalla suite.	

**Tabella 6-3 - Punti di forza e debolezza di Kettle**

## Report Designer

Report Designer è uno strumento di reportistica che permette di creare, modificare e pubblicare, nella piattaforma Pentaho, report relazionali e analitici esportando i dati in una vasta gamma di formati come PDF, Excel, HTML, Text, Rich-Text-File e XML e CSV.

In generale, nelle soluzioni Open source più comuni, il processo di reporting segue il modello schematizzato in Figura 6-4



**Figura 6-4 - Architettura del processo di reporting**

L'architettura mostrata in Figura 6-4 è composta delle seguenti componenti:

- Un Report Designer che definisce le specifiche dei report;
- Le specifiche del report incluse in un file XML;
- Un *report engine* per eseguire il *report* secondo le specifiche e rendere l'output in diversi formati;
- Definizione della connessione al database che può utilizzare un *middleware* standard come JDBC per connettersi a diverse fonti di dati.

In Pentaho Report Designer un report è formato da varie sezioni di diverso tipo, alcune di queste sono standard, altre sono flessibili e possono essere aggiunte o rimosse.

Nella Tabella 6-4 vengono rappresentati i punti di forza e di debolezza dello strumento.

Punti di forza	Punti di debolezza
Semplifica le attività di creazione dei report, fornendo tutti gli elementi solitamente utilizzati nei report.	Non è possibile creare campi calcolati partendo da <i>data mart</i> diversi.
Possibilità di creare condizioni per nascondere o evidenziare elementi del report.	Per interrogazioni particolarmente complesse, l'uso dello strumento SQL Query Designer può risultare insoddisfacente.
Accesso a fonti dati eterogenee e secondo vari formati.	L'uso di alcune funzioni può essere poco intuitivo.
Parametrizzazione dei report.	
Possibilità di esportare i report nei formati più comuni (html, excel, pdf, testo).	
Possibilità di creare sotto report.	
Possibilità di pianificare i report.	

**Tabella 6-4 - Punti di forza e debolezza di Report design**

## Schema Workbench

Lo schema Workbench offre un interfaccia grafica per definire, creare e testare gli schemi XML dei cubi creati con Mondrian.

Questi schemi possono utilizzare sia tabelle dei fatti esistenti, sia tabelle delle dimensioni trovate nel RDBMS.

Questo strumento permette in modo visuale di comporre gli schemi (risulta molto utile per chi non conosce la sintassi dei tag XML accettati da Mondrian).

Grazie alla semplicità d'uso dell'applicazione vengono ridotti gli errori, soprattutto in presenza di schemi particolarmente complessi.

Schema Workbench è provvisto delle seguenti funzionalità:

- editor di schemi integrato con il data source sottostante;
- *testing* delle query MDX.

Nella Tabella 6-5 vengono rappresentati i punti di forza e di debolezza dello strumento.

Punti di forza	Punti di debolezza
Facile da usare.	Interfaccia poco curata.

**Tabella 6-5 - Punti di forza e debolezza di Schema Workbench**

## 6.3 pgAdmin

Scritto in C++, pgAdmin grazie a una interfaccia grafica, consente di amministrare in modo semplificato *database* PostgreSQL.

Sia agli amministratori del database che gli utenti, posso utilizzare l'applicazione grazie alla presenza della gestione dei permessi prelevati direttamente dal database PostgreSQL.

pgAdmin permette di creare un *database* da zero, creare, modificare e cancellare le tabelle ed eseguire operazioni di ottimizzazione e modifica dei dati sulle stesse.

Grazie a una particolare funzione è in grado di effettuare il backup e il ripristino dei dati senza il bisogno di scrivere comandi SQL.

## **7 PROCEDURE PER LA REALIZZAZIONE DEI *DATA MART***

In questo capitolo verranno illustrate le fasi di estrazione, trasformazione e caricamento (ETL) effettuate al fine di rendere disponibili i dati nei *data mart* per la successiva generazione di report in modo da rispondere alle analisi precedentemente definite.

Per ogni dimensione e tabella dei fatti è stato creato un processo ETL. Inizialmente sono state create le trasformazioni per le dimensioni in modo da poter creare, successivamente, chiavi esterne surrogate nella tabella dei fatti. Le trasformazioni delle dimensioni sono: “dim\_agente”, “dim\_succursale”, “dim\_campagnapromozionale”, “dim\_tempo”, “dim\_cliente”, “dim\_prodotto”.

Nella seconda fase sono state create le trasformazioni delle *fact table*: “fact\_ordini” e “fact\_venduto”.

Nei paragrafi seguenti verrà mostrata solo una trasformazione, quella che riguarda il popolamento della *fact table* “fact\_ordini”, in quanto racchiude tutti i *task* utilizzati dalle restanti trasformazioni.

## 7.1 La fase di estrazione

I dati provengono da una fonte, quindi tutte le trasformazioni utilizzeranno la connessione al *database* di nome “Openbravo” (di tipo Postgres) nel task “Table input”.

Nella prima fase del processo di ETL, dopo aver indentificato le tabelle del *database* contenenti i dati necessari alle analisi (Figura 6.6), si procede alla loro estrazione in modo da renderli disponibili nello *stream*<sup>4</sup>.

Questo task permette di formulare ed eseguire una *query* SQL che estrae gli attributi necessari dalla tabella di origine e li rende disponibili nello *stream*.

Nella Figura 7-1 è illustrato il *task* utilizzato dalla trasformazione per il popolamento di *fact\_ordini* per l'estrazione dei dati dalla tabella *C\_ORDERLINE*.

Lo *stream*, dopo l'esecuzione del *task* “Table input”, avrà i seguenti attributi:

- **c\_orderline\_id:** campo dal quale è possibile ricavare la chiave primaria della riga dell'ordine;
- **ad\_org\_id:** chiave esterna per la tabella *AD\_ORG*;
- **c\_bpartner\_id:** chiave esterna per la tabella *C\_BPARTNER*;
- **c\_order\_id:** campo dal quale è possibile ricavare la chiave primaria dell'ordine;

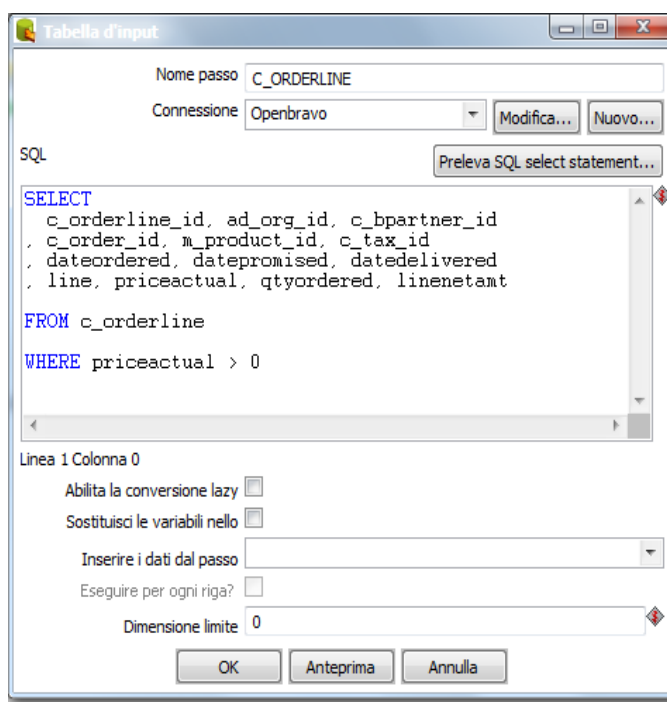


Figura 7-1 – Il *task* “Table Input”

<sup>4</sup> Flusso delle informazioni di un processo ETL

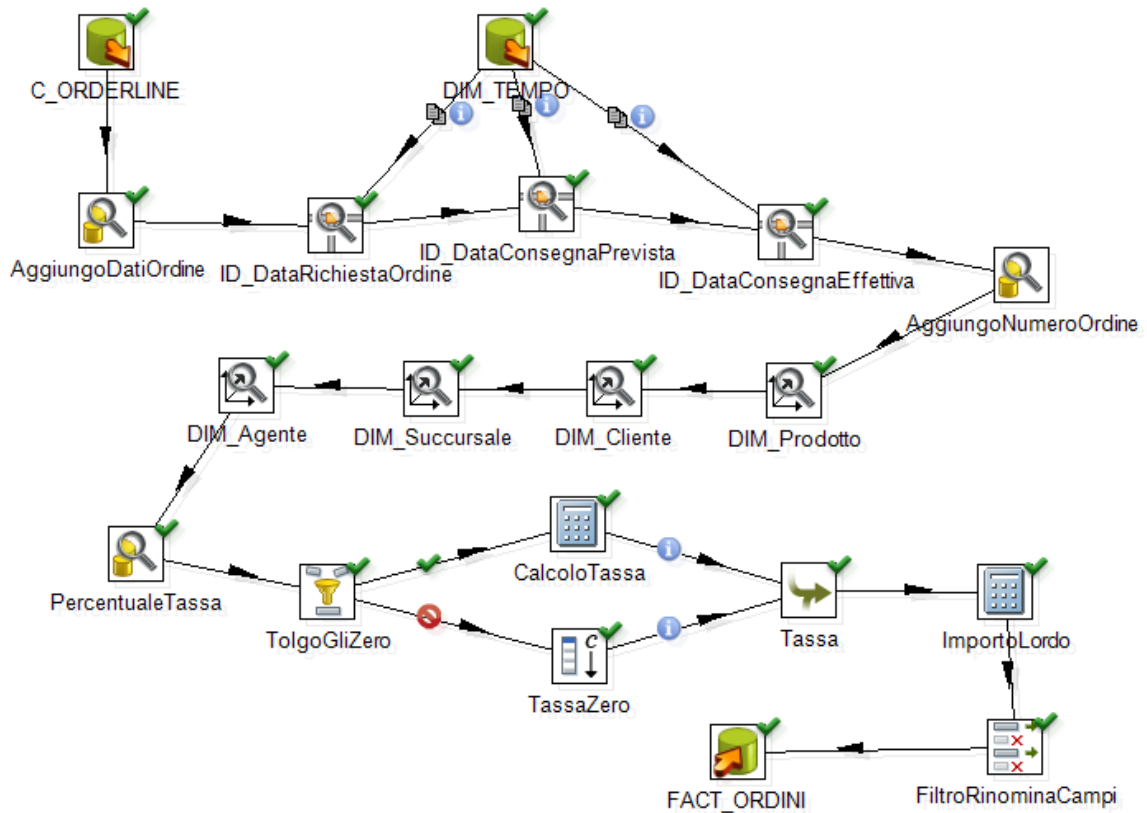


- **m\_product\_id:** chiave esterna per la tabella M\_PRODUCT;
- **c\_tax\_id:** chiave esterna per la tabella C\_TAX;
- **dateordered:** campo dal quale è possibile ricavare la data dell'ordine;
- **datepromised:** campo dal quale è possibile ricavare la data prevista di evasione dell'ordine;
- **datedelivered:** campo dal quale è possibile ricavare la data di spedizione dell'ordine;
- **priceactual:** campo dal quale è possibile ricavare il prezzo del prodotto ordinato;
- **qtyordered:** campo dal quale è possibile ricavare la quantità del prodotto ordinato;
- **linenetamt:** campo dal quale è possibile ricavare l'importo netto della riga di un determinato ordine;

## 7.2 La fase di trasformazione

La fase di trasformazione è di fondamentale importanza nel processo di ETL in quanto permette la trasformazione dei dati in ingresso in opportuni formati.

Nella Figura 7-2 è riportato uno *screen shot* dell'intero processo di trasformazione, dove i nomi degli *step* rappresentano i rispettivi task.



**Figura 7-2 – La trasformazione per il caricamento dei dati in fact\_ordini**

Dopo il caricamento dei dati si effettua, utilizzando il task “Database Lookup”, una giunzione fra lo *stream* della trasformazione (risultato della precedente *query* sulla tabella C\_ORDERLINE) e la tabella C\_ORDER con lo scopo di associare a ogni riga di un ordine, i dati della testata relativi a quell’ordine, Figura 7-3.

Nome del passo: AggiungoDatiTestata

Connessione: Openbravo [Modifica...] [Nuovo...]

Schema di lookup: [ ] [Apri]

Tabella di lookup: c\_order [Sfoggia...]

Abilitare cache? ☐

Dimensione cache in righe (0=tutto nella cache): 0

Caricare tutti i dati dalla tabella ☐

La chiave/i di lookup per il valore/i:

#	Campo della tabella	Comparatore	Campo1	Campo2	
1	c_order_id	=	c_order_id		

I valori da ritornare dalla tabella di lookup:

#	Campo	Nuovo nome	Default	Tipo	
1	salesrep_id		0	String	

Non passare la riga se il lookup fallisce ☐

Fallire sui risultati multipli? ☐

Ordinato per: [ ]

[OK] [Annulla] [Preleva campi] [Preleva campi di lookup]

**Figura 7-3 - Il task "Database Lookup"**

Questo task, dato un campo dello *stream* e un attributo della tabella di *lookup*, ricerca i valori in tutte le righe dell'attributo della tabella di *lookup*.

Come si evince dalla Figura 7-3, allo *stream* dei dati viene aggiunto l'attributo `salesresp_id` che rappresenta la chiave esterna verso la tabella `C_BPARTNER`.

Il passaggio successivo permette di convertire le date presenti nello *stream* in chiavi esterne in riferimento alla `DIM_TEMPO`. Per effettuare questa conversione si utilizza il task "Stream lookup", che effettua un *lookup* fra un campo dello *stream* della trasformazione (campo "dateordered") e l'attributo ("timedate") della tabella `DIM_TEMPO`.

Nome del passo

Passo di lookup

La chiave/i per il lookup del valore/i:

#	Campo	Campo lookup
1	dateordered	timedate

Specifica i campi da recuperare:

#	Campo	Nuovo nome	Default	Tipo
1	id_tempo	ID_DataRichiestaOrdine	0	Integer

Preserva la memoria (costa) ☒  
 Chiave e valore sono ☐  
 Utilizza lista ordinata (i.s.o.) ☒

**Figura 7-4 - Il task "Stream Lookup"**

Come si evince dalla Figura 7-4, allo *stream* viene aggiunto l'attributo "ID\_DateRichiestaOrdine" che rappresenta la chiave esterna verso la tabella DIM\_TEMPO.

Un altro task importante è il "Dim lookup" che, come il task precedente, effettua un *lookup* fra un campo dello *stream* della trasformazione (campo "id\_openbravo\_prodotto") e l'attributo ("m\_product\_id") della dimensione DIM\_PRODOTTO (Figura 7-5).

Lookup / Aggiornamento dimensione

Nome del passo DIM\_Prodotto

Aggiornare la dimensione? ☐

Connessione postgres Modifica... Nuovo...

Schema di destinazione Apri

Tabella di destinazione dim\_prodotto Sfogli...

Dimensione commit 100

Abilitare la cache? ☒

Pre-caricare la cache? ☐

Dimensione cache in righe (0=tutto nella cache) 5000

**Chiave** **Campi**

Campi chiave (per il lookup di riga nella dimensione):

#	Campo dimensione	Campo nello stream
1	id_openbravo_prodotto	m_product_id

Campo della chiave tecnica id\_prodotto Nuovo nome

Creazione della chiave tecnica

☒ Utilizza il massimo della tabella + 1

☐ Utilizza sequenza

☐ Utilizza campo di auto incremento

Campo di versione version

Campo data stream

Campo di inizio range data date\_from Anno min. 1900

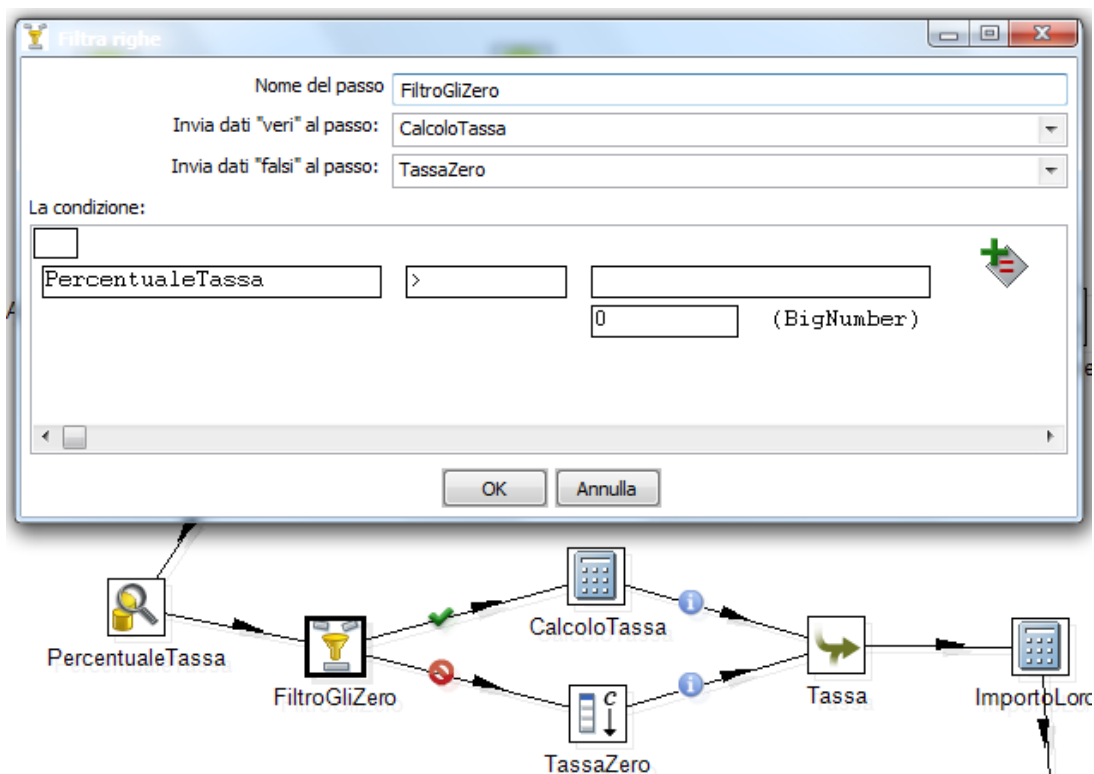
Utilizzare una data d'inizio alternativa? ☐ <Seleziona opzione>

Fine range data di tabella date\_to Anno max 2199

OK Annulla Preleva campi SQL

**Figura 7-5 - Il task "Dim Lookup"**

Proseguendo con la trasformazione, si hanno una serie di comandi degni di nota e riportati in Figura 7-6.



**Figura 7-6 - Serie di comandi per il calcolo della tassa di un prodotto**

Dopo aver inserito nello *stream* il campo “PercentualeTassa”, che rappresenta la percentuale di tassa a cui è soggetto il prodotto, viene creato un filtro con il task “Filter rows”.

Questo task permette di separare lo *stream* in due parti:

1. quella la cui condizione è vera (ha il simbolo di una V verde);
2. quella la cui condizione è falsa (ha il simbolo di un cerchio rosso con un taglio).

La prima parte dello *stream* viene inviata al task “CalcoloTassa” che, utilizzando una funzione aritmetica, converte la percentuale in un numero e inizializza l’attributo “Tassa”. La seconda parte viene inviata al task “TassaZero” che inizializza al valore zero l’attributo “Tassa”. Per riunire lo stream viene utilizzato il task “Append” che permette di unire due *stream* con gli stessi attributi.

Grazie alla moltitudine di comandi, Pentaho, è in grado di offrire allo sviluppatore molti modi per la creazione delle trasformazioni. Per far capire questo concetto verrà mostrato il task “Database Join” (Figura 7-7), presente nella trasformazione che realizza la DIM\_AGENTE.

Come si evince dalla Figura 7-7 Il task “Database Join” effettua una *Join* tra il campo “c\_bpartner\_id” dello *stream* e il campo “c\_bpartner\_id” della tabella “c\_bpartner\_location”, aggiungendo allo *stream* il campo “c\_location\_id”.

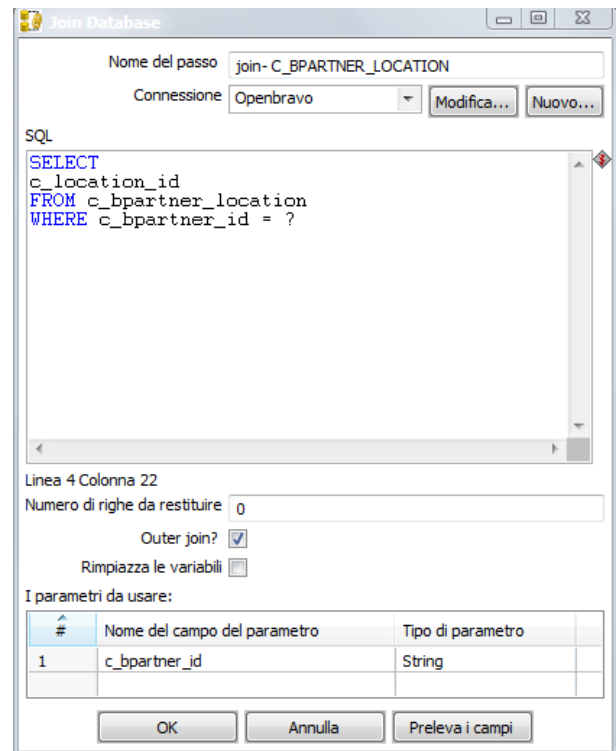


Figura 7-7 - Il task "Database Join"

Dalla descrizione del task, esso può sembrare simile al task “Database lookup”, ma sono diversi per via dell’algoritmo usato:

- “Database Join”, dati due o più campi da confrontare, prima li ordina in modo crescente e poi effettua la ricerca dei valori fra gli attributi;
- “Database lookup” non ordina i campi da confrontare, ma effettua subito la ricerca.

Ogni task ha i suoi pregi e difetti (Tabella 7-1), questo non permette di sapere in anticipo quale sia meglio utilizzare nella trasformazione, infatti la soluzione migliore è utilizzarli entrambi e scegliere quello più efficiente.

Database join	Database lookup
Con pochi record è più lento in quanto perde tempo ad ordinare i dati in memoria e se non c’è abbastanza spazio utilizza un file.	Molto veloce con pochi record.
Più veloce se la tabella principale ha molte chiavi esterne uguali verso la tabella di destinazione.	Più veloce se la tabella principale ha molte chiavi esterne diverse verso la tabella di destinazione.
Risulta essere più veloce con una grande mole di dati.	Molto lento con una grande mole di dati

Tabella 7-1- Pregi e difetti del task "Database Join" e del task "Database lookup"

## 7.3 Il processo di caricamento

Per quanto riguarda la fase di caricamento delle tabelle dimensionali, essa è avvenuta per mezzo del task “Dimension Lookup/Update” che permette la creazione della chiave surrogata per ogni tabella dimensionale e la creazione nello *stream* di dati della chiave esterna per la tabella dei fatti.

In Figura 7-8 è mostrato tale *step* relativamente alla dimensione DIM\_AGENTE.



Lookup / Aggiornamento dimensione

Nome del passo DIM\_AGENTE

Aggiornare la dimensione? ☒

Connessione postgres

Schema di destinazione

Tabella di destinazione DIM\_AGENTE

Dimensione commit 100

Abilitare la cache? ☐

Pre-caricare la cache? ☐

Dimensione cache in righe (0=tutto nella cache)

Chiave Campi

Campi di lookup/update

#	Campo dimensione	Campo di stream con cui comparare	Tipo di aggiornamento della dimensione
1	NomeCognomeAgente	name	Inserisci
2	CittaAgente	Citta	Inserisci
3	AreaGeograficaAgente	AreaGeografica	Inserisci

Campo della chiave tecnica ID\_Agente Nuovo nome

Creazione della chiave tecnica

☒ Utilizza il massimo della tabella + 1

☐ Utilizza sequenza

☐ Utilizza campo di auto incremento

Campo di versione version

Campo data stream

Campo di inizio range data date\_from Anno min. 1930

Utilizzare una data d'inizio alternativa? ☐ <Seleziona opzione>

Fine range data di tabella date\_to Anno max 2099

OK Annulla Preleva campi SQL

Figura 7-8- Il task "Dimension Lookup/Update"

Questo task, dopo aver indicato i campi di *lookup* e *update* e il tipo di aggiornamento che si vuole avere nella dimensione (in questo esempio è di Tipo 2), questo task aggiunge tre campi, "version", "date\_from", "date\_to" alla tabella della

dimensione, che rappresentano: la versione, data d'inizio e data di fine della versione del record della dimensione.

Considerando invece il caricamento della *fact table*, essa è stata condotta per mezzo del task "Table Output" che permette la scrittura degli attributi presenti nello *stream* nella tabella dei fatti.

Dopo aver opportunamente rinominato ed eliminato gli attributi che non contengono informazioni utili tramite lo *step* "FiltroRinominaCampi" (Figura 7-9) vengono caricati i dati con il task "Table output" (Figura 7-10).

Dopo aver impostato la connessione al *database* e la tabella di destinazione ("fact\_ordini"), viene caricato lo *stream*.

A questo punto è conclusa la fase di realizzazione del *data mart*, e si procederà con la definizione dello schema virtuale all'interno del *server OLAP Mondrian* e la definizione delle *query* per la realizzazione dei report per le analisi.



Figura 7-9 - Comandi per il processo di caricamento delle *fact table*.

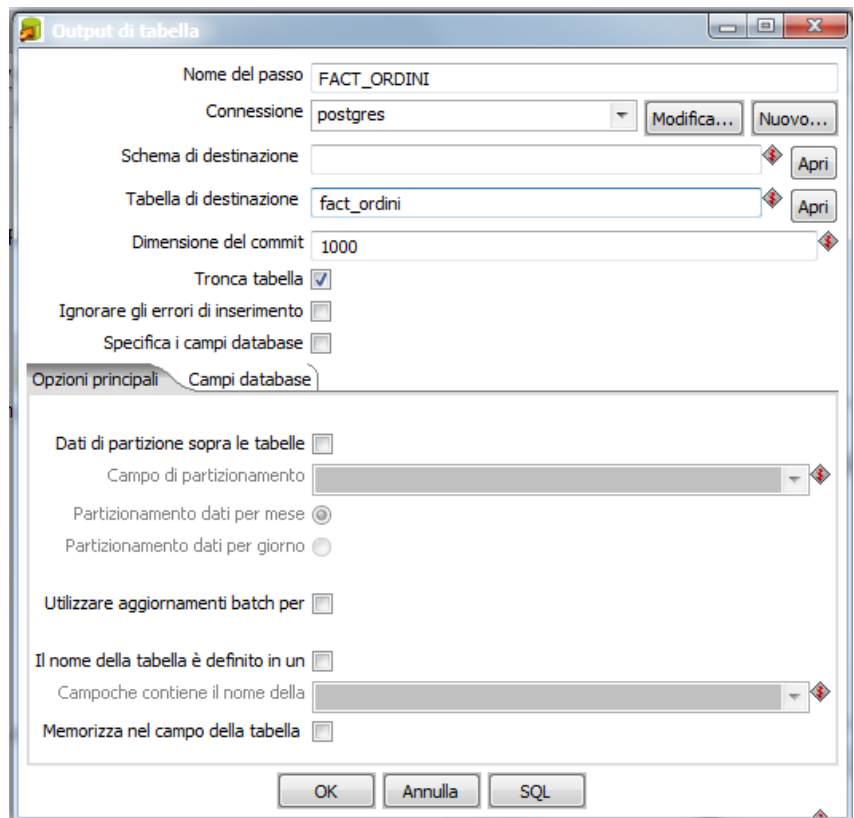


Figura 7-10 - Il task "Table output"

## 8 IL MODELLO MULTIDIMENSIONALE

In questo capitolo, dopo aver introdotto alcune parole chiave della tecnologia OLAP, si procederà alla progettazione dello schema OLAP e alla sua definizione.

### 8.1 La tecnologia OLAP

Dopo aver creato e caricato i *data mart* con i dati provenienti dalla base di dati operativa, si passa alla definizione delle fasi di analisi che permetteranno la

formulazione di report. Come anticipato nel Paragrafo 5.1.1, verrà utilizzata la tecnologia OLAP per l'interrogazione e visualizzazione dei dati.

Per caratterizzare questa tecnologia viene utilizzato il modello FASMI (*Fast Analysis of Shared Multidimensional Information*)<sup>5</sup> come segue [Alb12]:

1. **Veloce (Fast).** Tali sistemi, usati di solito in modo interattivo, devono fornire rapidamente i risultati (generalmente in alcuni secondi e raramente in più di 20 o 30 secondi). Questo livello di prestazioni è cruciale per invogliare gli analisti a lavorare efficacemente sui dati.
2. **Analisi (Analysis).** Tali sistemi devono fornire un ampio repertorio di funzioni analitiche riducendo al minimo la necessità di doverle definire con opportuni programmi.
3. **Condiviso (Shared).** Di solito un sistema OLAP è una risorsa condivisa e quindi devono essere previsti opportuni meccanismi di controllo degli accessi ai dati.
4. **Multidimensionale (Multidimensional).** Un requisito fondamentale dei sistemi OLAP è la visione multidimensionale dei dati con la possibilità di cambiare rapidamente le prospettive di analisi e i livelli di dettaglio, sfruttando la presenza di gerarchie.
5. **Informazione (Information).** I sistemi OLAP sono progettati per gestire grandi quantità di dati e per consentire loro analisi di varia natura per produrre utili informazioni sintetiche rappresentate in modi diversi (tabellare o grafica). I dati dei sistemi OLAP di solito provengono da più basi di dati operazionali e da fonti esterne.

---

<sup>5</sup> Termine coniato da Nigel Pendse autore nel 2004 del documento "The OLAP Report", nella pagina web "What Is OLAP?", dove dice che il concetto OLAP può essere definito in 5 regole.

In un modello OLAP, le informazioni sono gestite concettualmente come cubi, composti da categorie descrittive (dimensioni) e valori quantitativi (misure). In Figura 8-1 è fornito un esempio di cubo multidimensionale relativo al *data mart* progettato per il fatto “Ordini di vendita”. Per semplicità grafica sono state inserite solo tre dimensioni di analisi (Prodotto, Data e Succursale) e come misura il valore dell’ImportoNetto.

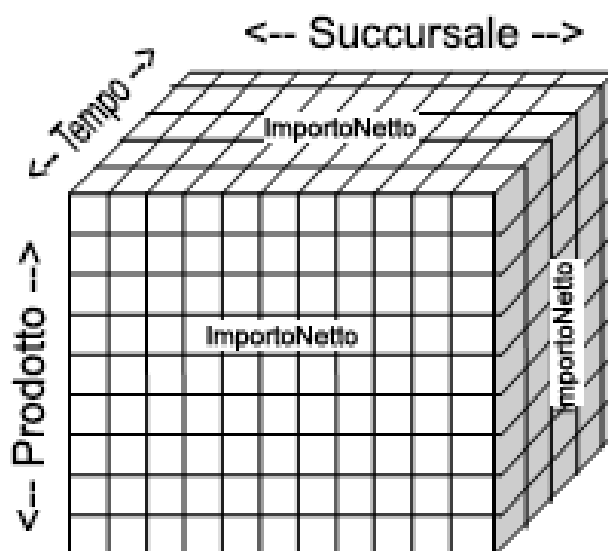


Figura 8-1 - Esempio di Cubo OLAP

All'interno di ogni dimensione di un modello OLAP, i dati sono organizzati in gerarchie per rappresentare i diversi livelli di dettaglio. Ad esempio, considerando la dimensione Prodotto esisteranno i livelli MacroCategoria, Categoria e Prodotto.

I cubi, le dimensioni, le gerarchie e le misure sono gli elementi essenziali dell'esplorazione multidimensionale OLAP, che può essere compiuta attraverso le funzioni base (Figura 8-2), quali:

- *Slicing*. Seleziona una fetta del cubo, con una restrizione su una dimensione;
- *Dicing*. Seleziona un sottocubo, con una restrizione su due o più dimensioni;
- *Drill-down*. Aumenta il livello di dettaglio dell'analisi, considerando più dimensioni o aggregando per attributi dimensionali più specifici;
- *Roll-up*. Diminuisce il livello di dettaglio delle analisi, considerando meno dimensioni;

- *Pivot*. Consente di ottenere rappresentazioni alternative dei dati ruotando il cubo.

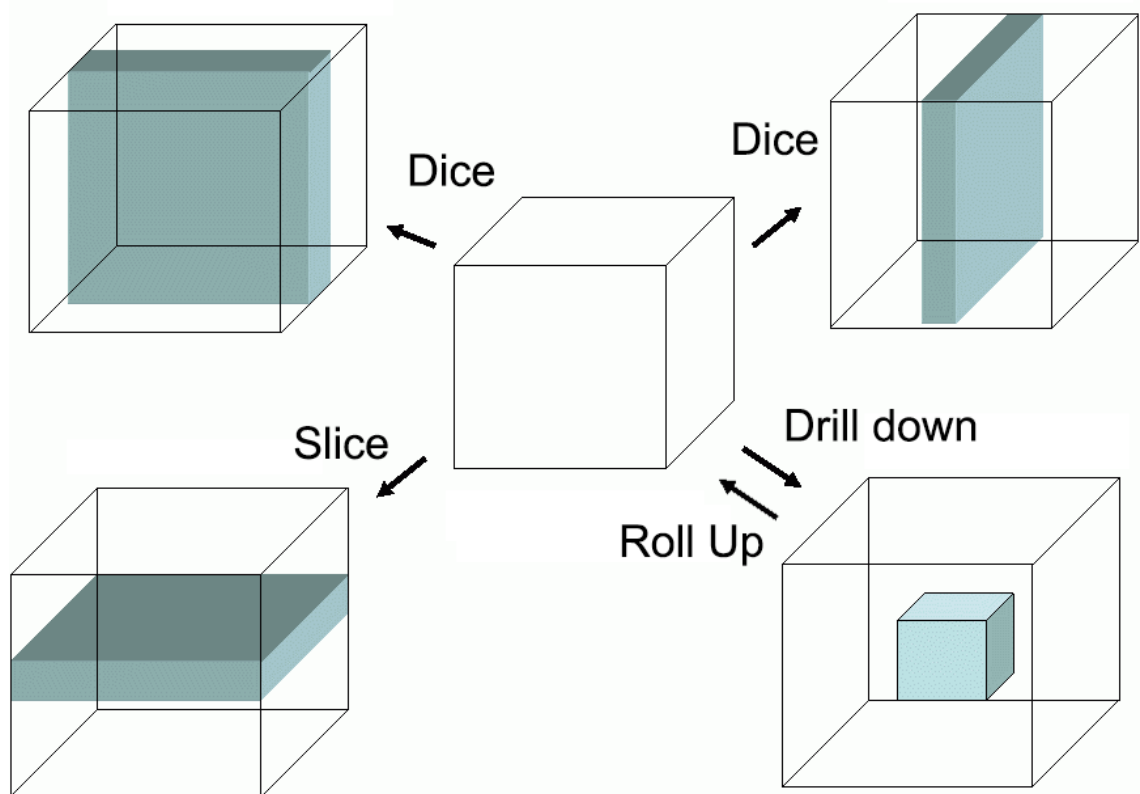


Figura 8-2 - Funzioni base di un cubo OLAP

## 8.2 Progettazione dello schema OLAP

Per la creazione e progettazione dello schema OLAP verrà utilizzata l'applicazione Schema Workbench, nella quale si potrà definire il *database* multidimensionale contenente il modello logico costituito dal cubo, gerarchie e membri, e le relazioni tra il modello multidimensionale e quello relazionale.

La progettazione dello schema OLAP prevede la definizione di ogni singolo elemento che costituisce il cubo per le analisi. Dopo aver stabilito la connessione con il *database* occorre specificare:

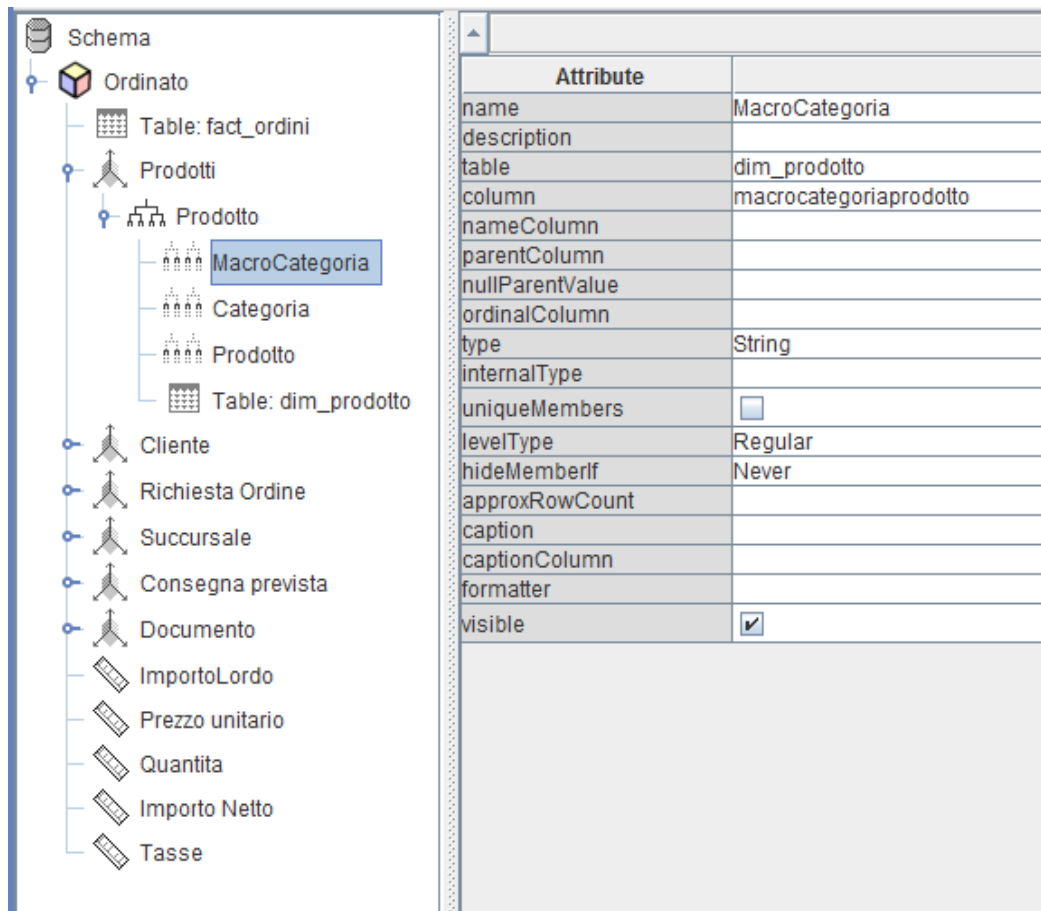
- **la tabella dei fatti**: contiene le chiavi surrogate (che fungono da chiavi esterne alle dimensioni), le misure e le dimensioni degeneri;

- **le tabelle dimensionali:** vengono impostate le diverse gerarchie e i relativi livelli di aggregazione. Per ogni gerarchia va specificata la chiave primaria a cui fare riferimento nella tabella (generata in precedenza con Spoon). A ogni livello corrisponde un diverso attributo all'interno della tabella dimensionale associata. Per le dimensioni degeneri non va impostata alcuna tabella dimensionale in quanto viene presa di default la tabella dei fatti.
- **le misure:** corrispondono ad una singola colonna nella tabella dei fatti e per ognuna di esse va specificata la modalità di aggregazione (somma, media, minimo, massimo, ecc). E' possibile aggiungere più volte una stessa misura e impostare una modalità di aggregazione differente (metrica).

La Figura 8-3 mostra come risulta la definizione finale dello schema OLAP tramite Schema Workbench. L'esempio in considerazione è quello riferito al fatto "Ordini di Vendita".

Sulla parte sinistra viene mostrato l'albero con le varie componenti del cubo: la tabella dei fatti, le dimensioni e le misure.

La dimensione Prodotti è espansa ed è caratterizzata da un'unica gerarchia chiamata Prodotto la quale, a sua volta, è costituita dai tre livelli MacroCategoria, Categoria e Prodotto. La tabella dimensionale associata è la tabella dim\_prodotti presente nel *data warehouse*.



**Figura 8-3 - Definizione Schema OLAP con Schema Workbench (Ordini di Vendita)**

Sulla parte destra vengono visualizzate le proprietà da impostare. Per ogni livello di una gerarchia vanno definiti: il nome del livello (*name*), la tabella dimensionale (*table*), la relativa colonna (*column*), il tipo di dato (*type*) e il tipo di livello (*levelType*).

### 8.3 Definizione schema XML Mondrian

Il passo preliminare alla realizzazione delle analisi è la definizione dello schema Mondrian in formato XML, il quale definisce il *database* multidimensionale contenente il modello logico costituito dal cubo, gerarchie e membri, e le relazioni tra il modello multidimensionale e quello relazionale.



Parte del codice XML generato da Schema Workbench è il seguente<sup>6</sup>:

```
<-- DICHIARAZIONE SCHEMA-->
<Schema name="ordini">

  <-- DICHIARAZIONE CUBO -->
    <Cube name="Ordinato" visible="true" cache="true"
enabled="true">
      <Table name="fact_ordini" schema="public"/>

  <-- DICHIARAZIONE DIMENSIONE PRODOTTO -->
  <Dimension type="StandardDimension" visible="true"
foreignKey="id_prodotto" highCardinality="false" name="Prodotti">

    <-- DICHIARAZIONE DELLA GERARCHIA -->
    <Hierarchy name="Prodotto" visible="true" hasAll="true"
allMemberName="Tutti i Prodotti" primaryKey="id_prodotto">
      <Table name="dim_prodotto" schema="public" />

      <Level name="MacroCategoria" visible="true"
table="dim_prodotto" column="macrocategoriaprodotto" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never" />
      <Level name="Categoria" visible="true"
table="dim_prodotto" column="categoriaprodotto" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never" />
      <Level name="Prodotto" visible="true" table="dim_prodotto"
column="siglaprodotto" nameColumn="descrizionecommercialeprodotto"
type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never" />
    </Hierarchy>

  </Dimension>

  <-- DICHIARAZIONE DIMENSIONE DEGENERARE DOCUMENTO -->
  <Dimension type="StandardDimension" visible="true"
highCardinality="false" name="Documento">
    <Hierarchy name="Ordine" visible="true" hasAll="true">
      <Level name="Ordine" visible="true" column="numeroordine"
type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never" />
    </Hierarchy>
  </Dimension>

  ...
  ...
  ...

  <-- DICHIARAZIONE DELLE MISURE -->
  <Measure name="ImportoLordo" column="importolordo"
datatype="Integer" formatString="#&8364; #,###.00" aggregator="sum"
visible="true" />
```

---

<sup>6</sup> Relativo allo schema OLAP per il fatto "Ordini di vendita".

```

<Measure name="Quantita" column="quantita" datatype="Numeric"
formatString="#,###" aggregator="sum" visible="true" />

<Measure name="Importo Netto" column="importonetto"
datatype="Integer" formatString="&#8364; #,###.00" aggregator="sum"
visible="true" />

<Measure name="Tasse" column="tasse" datatype="Integer"
formatString="&#8364; #,###.00" aggregator="sum" visible="true" />

<-- TAG DI CHIUSURA -->

    </Cube>
</Schema>

```

Nella parte iniziale dello schema avviene la dichiarazione del cubo Ordini relativo al fatto “Ordini di vendita”. Per maggiore leggibilità è stata trascritta solo la dichiarazione della dimensione Prodotto (in quanto vi è presente una gerarchia) e la dimensione degenerare Documento.

La parte finale dello schema presenta la dichiarazione delle misure ImportoLordo, Quantita, ImportoNetto e tasse con le relative funzioni di aggregazione.

## 8.4 Pubblicazione sul server Pentaho

Una volta terminata la definizione dello schema OLAP, e controllato la sua correttezza, viene pubblicato sul *server* di Pentaho per renderlo disponibile per le analisi.

Prima viene effettuata la connessione al *repository* del *server* e successivamente salvato il file specificando la connessione associata. La connessione al *data warehouse* deve essere aggiunta a priori tra le connessioni rese disponibili al *server* di Pentaho tramite la Pentaho Administrator Console (PAC) o tramite la Pentaho User Console (PUC).

La connessione al *repository* di Pentaho e la pubblicazione dello schema XML è mostrata in Figura 8-4.

The image shows two overlapping Windows-style dialog boxes. The left dialog, titled 'Repository Login', contains fields for 'Server URL' (set to 'http://localhost:8080/pentaho/'), 'Publish Password' (masked with dots), 'Pentaho Credentials' with 'User' (set to 'joe') and 'Password' (masked), and a checked 'Remember these Settings' checkbox. The right dialog, titled 'Publish Schema', contains 'Schema Name' (set to 'ordini2'), 'Schema File' (set to 'ordini.mondrian.xml'), and 'Location' (set to '/ordini'). It features a table with columns 'Name', 'Type', and 'Date Modified'. Below the table is a 'Publish Settings' section with 'Pentaho or JNDI Data Source' (set to 'DatiOrdini') and an unchecked 'Register XMLA Data Source' checkbox. Both dialogs have 'OK' and 'Cancel' buttons.

Name	Type	Date Modified
------	------	---------------

Figura 8-4 - Connessione al repository del server Pentaho e pubblicazione dello schema XML

## **9 DEFINIZIONE DEI REPORT**

Completata la definizione dello schema XML si passa alla formulazione delle query in formato MDX per esplicitare le particolari analisi richieste, al fine di produrre i report desiderati.

## 9.1 Analisi Ordini di vendita

### Ordinato totale per agente e sede “Succursale”.

Considerando il primo requisito di analisi richiesto il codice della formulazione della query MDX è riportato nel Listato 9-1.

```
SELECT  
NON EMPTY {Hierarchize({[Measures].[Importo Netto]})} ON COLUMNS,  
NON EMPTY CrossJoin([Succursale].[Succursale].Members, [Agente].[Agente].Members)  
ON ROWS  
FROM [Ordinato]
```

Listato 9-1 - Ordinato totale per agente e sede “Succursale”

Come di evince dal Listato 9-1, nella clausola SELECT è stata specificata la misura [Measures].[Importo Netto] da visualizzare sull’asse verticale (ON COLUMNS), mentre sull’asse orizzontale (ON ROWS) vengono visualizzate le sedi “Succursale” ([Succursale].[Succursale].Members) e gli agenti ([Agente].[Agente].Members).

Per comprendere meglio la query MDX bisogna spiegare le funzioni:

- **NON EMPTY.** Restituisce il set delle tuple non vuote da un set specificato, in base al prodotto incrociato tra il set specificato e un secondo set.
- **Hierarchize().** Organizza i membri del set specificato in ordine gerarchico e mantiene sempre i duplicati.
- **CrossJoin().** Restituisce il prodotto incrociato di uno o più set.

In Figura 9-1 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Succursale	Agente	Importo Netto
F&B España - Región Norte	Giuseppe López	€ 1.762.459,00
	Javier Martín	€ 1.231.838,00
	Joana Russo	€ 2.606.412,80
F&B España - Región Sur	Antonio Rossi	€ 1.650.223,00
	Leandro Giannotti	€ 575.010,00
	Rocco Amato	€ 1.898.280,00
	John Smith	€ 3.033.947,50

Figura 9-1 - Ordinatio totale per agente e sede “Succursale”

**Rango dei clienti e delle “aree geografiche” in termini di volume degli ordini nell’ultimo anno.**

Considerato il requisito di analisi richiesto, il codice della formulazione della query MDX è riportato nel Listato 9-2.

```

With
Set [*FILTRO_BASE] as 'Filter(NonEmptyCrossJoin([Cliente].[Cliente].Members, [Richiesta Ordine].CurrentMember), Not IsEmpty ([Measures].[Importo Netto]))'

Member [Measures].[Rango] as 'Rank((([Cliente].currentMember),
Generate(Exists([*FILTRO_BASE], {(Ancestor([Cliente].CurrentMember, [Cliente].[Area Geografica Cliente]))}), {[Cliente].currentMember})), [Measures].[Importo Netto])'

Select
{[Measures].[Importo Netto],[Measures].[Rango]} ON COLUMNS,

Hierarchize({[Cliente].[Area Geografica Cliente].Members,
Order(Generate([*FILTRO_BASE], {[Cliente].currentMember})), [Measures].[Rango], ASC) })
ON ROWS

From [Ordinato]
Where (Generate([*FILTRO_BASE], {[Richiesta Ordine].currentMember}))

```

**Listato 9-2** - Rango dei clienti e delle “aree geografiche” in termini di volume degli ordini nell’ultimo anno

Come si evince dal Listato 9-2, viene creato un filtro, chiamato [\*FILTRO\_BASE], utilizzando il comando **Filter()**<sup>7</sup> a cui viene dato come input una *join* creata con il

<sup>7</sup> Restituisce il set risultante dal filtro di un set specificato in base a una condizione di ricerca.

comando **NonEmptyCrossJoin()**<sup>8</sup> tra il Cliente, l'anno attuale e la misura ImportoNetto.

Successivamente viene creata la misura che calcola il rango ([Measures].[Rango]) per AreaGeografica e Cliente utilizzando il comando **Rank()**<sup>9</sup> a cui viene dato in input il [\*FILTRO\_BASE] e il comando **Ancestor()**<sup>10</sup> che restituisce il predecessore di un membro specificato al livello specificato e la misura "ImportoNetto".

Nella clausola SELECT sono state specificate le misure [Measures].[Importo Netto] e [Measures].[Rango] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzate le AreeGeografiche, il Cliente e il Rango.

In Figura 9-2 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Area Geografica Cliente	Cliente	Importo Netto	Rango
Spain		€ 47.210.102,86	1
	Bebidas Alegres, S.L.	€ 20.546.130,00	1
	Refrescos Naturales, S.A.	€ 10.271.300,00	2
	La Fruta es la Vida, S.L.	€ 5.422.150,00	3
	Restaurantes Luna Llena, S.A.	€ 3.784.010,50	4
	Alimentos y Supermercados, S.A.	€ 3.763.055,56	5
	Hoteles Buenas Noches, S.A.	€ 3.392.958,30	6
	Vendor Technology	€ 12.825,00	7

Figura 9-2 - Rango dei clienti e delle "aree geografiche" in termini di volume degli ordini nell'ultimo anno.

**I 10 prodotti ordinati in un certo mese con la quantità più alta, per categoria di prodotto.**

<sup>8</sup> Restituisce un set che contiene il prodotto incrociato di uno o più set, escludendo le tuple vuote e le tuple a cui non sono associati dati di tabelle dei fatti.

<sup>9</sup> Restituisce il rango in base uno di una tupla specificata in un set specificato.

<sup>10</sup> Funzione che restituisce il predecessore di un membro specificato al livello specificato oppure alla distanza specificata dal membro.

Considerato il requisito di analisi richiesto, il codice della formulazione della query MDX è riportato nel Listato 9-3.

```
With
Set [*FILTRO_BASE] as 'NonEmptyCrossJoin([Prodotti].[Prodotto].Members,Filter([Richiesta
Ordine].[Mese].Members,([Richiesta Ordine].CurrentMember In {[Richiesta
Ordine].[2011].[2].[3].[LUG]})))'

Member [Measures].[*TOP_Quantity] as 'Rank([Prodotti].CurrentMember,
Order( Generate( [Prodotti].[Categoria].CurrentMember,{[Prodotti].CurrentMember})
,([Measures].[Quantita],[Richiesta Ordine].[*CTX_MEMBER_SEL~AGG])
,BDESC))'

Member [Richiesta Ordine].[*CTX_MEMBER_SEL~AGG] as
'Aggregate(Generate([*FILTRO_BASE], {[Richiesta Ordine].CurrentMember}))'

Select
{[Measures].[Quantita]} on columns, Hierarchize({[Prodotti].[Categoria].Members,
Order(Generate(Filter([*FILTRO_BASE],[Measures].[*TOP_Quantity] <= 10),
{([Prodotti].currentMember))}
, [Prodotti].CurrentMember,[Measures].[Quantita] ,ASC))} on rows

From [Ordinato]
WHERE {Hierarchize({[Richiesta Ordine].[2011].[2].[3].[LUG]}))}
```

**Listato 9-3 - I 10 prodotti ordinati in un certo mese con la quantità più alta, per categoria di prodotto.**

Come si evince dal Listato 9-3, viene creato un filtro, chiamato [\*FILTRO\_BASE], utilizzando il comando **Filter()** a cui viene dato come input una *join* creata con il comando **NonEmptyCrossJoin()** tra il prodotto e il mese in cui è avvenuta la richiesta dell'ordine.

Successivamente viene creata la misura [Measures].[\*TOP\_Quantity] che ha come risultato il rango (per quantità), per categoria di prodotto, dei prodotti presenti negli ordini di vendita.

Per calcolare la quantità totale dei prodotti per ciascuna categoria ([Richiesta Ordine].[\*CTX\_MEMBER\_SEL~AGG]) è stata usata la funzione **Aggregate()**<sup>11</sup>,

---

<sup>11</sup> Restituisce un numero che viene calcolato mediante aggregazione sulle celle restituite dall'espressione set.



inserendo come input la funzione **Generate()**<sup>12</sup> con valori [\*FILTRO\_BASE] e la richiesta dell'ordine.

Nella clausola SELECT è stata specificata la misura [Measures].[Quantita] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzate le categorie dei prodotti con i relativi 10 prodotti maggiormente presenti negli ordini di vendita.

In Figura 9-3 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Categoria	Prodotto	Quantita
Alcoholic		258.800
	Cerveza Lager 0,5L	70.000
	Cerveza Ale 0,5L	69.000
	Vino Rosado 0,75L	45.000
	Vino Tinto 0,75L	42.500
	Vino Blanco 0,75L	32.300

Figura 9-3 - I 10 prodotti ordinati in un certo mese con la quantità più alta, per categoria di prodotto.

### Tempi medi di attesa per evasione degli ordini.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-4.

Come di evince dal Listato 9-4, nella clausola SELECT è stata specificata la misura [Measures].[GiorniEvasione] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzate le Aeree Geografiche, le città e i clienti.

---

<sup>12</sup> Applica un set a ogni membro di un altro set e unisce i set risultanti tramite un join di unione. In alternativa, restituisce una stringa concatenata tramite la valutazione di un'espressione stringa su un set.

```

SELECT
NON EMPTY {Hierarchize({[Measures].[GiorniEvasione]})} ON COLUMNS,
NON EMPTY {Hierarchize({[Cliente].[Area Geografica Cliente].Members},
{[Cliente].[Città].Members}, {[Cliente].[Cliente].Members})} ON ROWS
FROM [Ordinato]

```

**Listato 9-4 - Tempi medi di attesa per evasione degli ordini**

In Figura 9-4 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Area Geografica Cliente	Città	Cliente	GiorniEvasione
Spain	La Coruña		12
		Alimentos y Supermercados, S.A	12
	Pamplona		113
		Customer A	105
		Customer B	194
	Sevilla		11
		Restaurantes Luna Llena, S.A.	11
	Valencia		11
		Hoteles Buenas Noches, S.A.	11
	United States		21

**Figura 9-4 - Tempi medi di attesa per evasione degli ordini**

## 9.2 Analisi Venduto

### Fatturato totale per agente e sede “Succursale”.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-5.

```

SELECT
NON EMPTY {Hierarchize({[Measures].[ImportoNetto]})} ON COLUMNS,
NON EMPTY CrossJoin([Succursale.SuccursaleGerarchia].[Succursale].Members,
[Agenti].[Agente].Members) ON ROWS
FROM [Venduto]

```

**Listato 9-5 - Fatturato totale per agente e sede “Succursale”.**

Come di evince dal Listato 9-5, nella clausola SELECT è stata specificata la misura [Measures].[ImportoNetto] da visualizzare sull’asse verticale (ON COLUMNS), mentre sull’asse orizzontale (ON ROWS) viene fatta un **CrossJoin()** tra gli agenti e le loro sedi “Succursali”.

In Figura 9-5 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Succursale	Agente	ImportoNetto
F&B España - Región Norte	Giuseppe López	€ 4.343.330,00
	Andrea Romeo	€ 6.533.510,00
	Massino Antonini	€ 606.200,00
	Joana Russo	€ 1.601.279,00
	Giovanni Putto	€ 1.210.620,00
	Rocco Amato	€ 3.232.400,00
	John Smith	€ 1.119.290,00
	Luciano Rismondi	€ 2.314.430,00
F&B España - Región Sur	Antonio Rossi	€ 1.765.100,00
	Federica Giusti	€ 64.299,00
	Leandro Giannotti	€ 33.900,00

**Figura 9-5 - Fatturato totale per agente e sede “Succursale”.**

#### **Rango degli agenti in termini di volume del venduto per mese.**

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-6.

```

With
Set [*FILTRO_BASE] as 'Filter(NonEmptyCrossJoin([Agenti].[Agente].Members,
[DataEmissione].[Mese].CurrentMember), Not IsEmpty ([Measures].[ImportoNetto]))'

Member [Measures].[Rango] as 'Rank(([Agenti].currentMember),
Generate(Exists([*FILTRO_BASE], {(Ancestor([Agenti].CurrentMember,
[Agenti].[Citta]))}), {[Agenti].currentMember})),[Measures].[ImportoNetto])'

Select
{[Measures].[ImportoNetto],[Measures].[Rango]} on columns,

Hierarchize({[Agenti].[Citta].Members,
Order(Generate([*FILTRO_BASE], {[Agenti].currentMember})), [Measures].[Rango],
ASC) }) on rows

From [Venduto]

```

**Listato 9-6 - Rango degli agenti in termini di volume del venduto per mese.**

Come si evince dal Listato 9-6, viene creato un filtro, chiamato [\*FILTRO\_BASE], utilizzando il comando **Filter()** a cui viene dato come input una *join* creata con il comando **NonEmptyCrossJoin()** tra gli agenti, il mese attuale e la misura ImportoNetto.

Successivamente viene creata la misura che calcola il rango ([Measures].[Rango]) per agente e città. Viene utilizzato il comando **Rank()** con in input il [\*FILTRO\_BASE] e il comando **Ancestor()** con input la CittaAgente e la misura "ImportoNetto".

Nella clausola SELECT sono state specificate le misure [Measures].[ImportoNetto] e [Measures].[Rango] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzate le Città e gli agenti ordinati per Rango.

In Figura 9-6 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Citta	Agente	ImportoNetto	Rango
Barcelona		€ 5.227.316,00	1
	Giuseppe López	€ 5.227.316,00	1
Pamplona		€ 12.674.352,40	1
	Andrea Romeo	€ 6.625.912,40	1
	Antonio Rossi	€ 3.386.715,00	2
	Federica Giusti	€ 2.661.725,00	3
San Sebastián		€ 3.376.383,70	1
	Leandro Giannotti	€ 2.066.273,50	1
	Massino Antonini	€ 1.310.110,20	2

Figura 9-6 - Rango degli agenti in termini di volume del venduto per mese.

**I 10 prodotti venduti in un certo mese con il volume di vendita più alto, per categoria di prodotto.**

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-7.

```

With
Set [*FILTRO_BASE] as
'NonEmptyCrossJoin([Prodotti].[Prodotto].Members,Filter([DataEmissione].[Mese].Members,([DataEmissione].CurrentMember In [DataEmissione].[2011].[02]])))'

Member [Measures].[*TOP_Quantity] as 'Rank([Prodotti].CurrentMember,
Order(Generate([Prodotti].[Categoria].CurrentMember,{[Prodotti].CurrentMember})
,[Measures].[ImportoNetto],[DataEmissione].[*CTX_MEMBER_SEL~AGG])
,BDESC))'

Member [DataEmissione].[*CTX_MEMBER_SEL~AGG] as
'Aggregate(Generate([*FILTRO_BASE],[DataEmissione].CurrentMember))'

Select
Hierarchize([Measures].[ImportoNetto]) ON COLUMNS,

Hierarchize([Prodotti].[Categoria].Members,
Order(Generate(Filter([*FILTRO_BASE],[Measures].[*TOP_Quantity] <= 10),
{[Prodotti].currentMember}))
,[Prodotti].CurrentMember,[Measures].[ImportoNetto],ASC)) on rows

From [Venduto]

```

Listato 9-7 - I 10 prodotti venduti in un certo mese con il volume di vendita più alto, per categoria di prodotto.

Come si evince dal Listato 9-7, viene creato un filtro, chiamato [\*FILTRO\_BASE], utilizzando la funzione **NonEmptyCrossJoin()** tra un prodotto e il mese in cui è avvenuta la vendita.

Successivamente viene creata la misura [Measures].[\*TOP\_Quantity] che ha come risultato il rango, dato un determinato mese, dei prodotti venduti divisi per categoria.

Per calcolare l'importo netto dei prodotti venduti per ciascuna categoria ([DataEmissione].[\*CTX\_MEMBER\_SEL~AGG]) è stata usata la funzione **Aggregate()** con input la funzione **Generate()** con valori [\*FILTRO\_BASE] e la data di emissione.

Nella clausola SELECT è stata specificata la misura [Measures].[ImportoNetto] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzati i 10 prodotti che hanno venduto maggiormente, divisi in categorie.

In Figura 9-7 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Categoria	Prodotto	ImportoNetto
Alcoholic		€ 19.090.516,40
	Vino Tinto 0,75L	€ 4.625.740,00
	Vino Blanco 0,75L	€ 4.000.165,00
	Cerveza Ale 0,5L	€ 2.939.640,00
	Vino Rosado 0,75L	€ 2.786.850,00
	Cerveza Lager 0,5L	€ 1.921.920,00
	Red Wine	€ 1.337.994,00
	Lager Beer	€ 640.886,40
	White wine	€ 457.314,00
	Ale Beer	€ 320.892,00
Bio	Rose wine	€ 59.115,00
		€ 3.374.790,00
	Orange Juice bio	€ 261.970,00
	Zumo de Naranja Bio 0.33l	€ 3.112.820,00

Figura 9-7 - I 10 prodotti venduti in un certo mese con il volume di vendita più alto, per categoria di prodotto.

### Tempi attesa medi di saldo delle fatture.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-8.

```
SELECT  
NON EMPTY {Hierarchize({[Measures].[GiorniPagamento]})} ON COLUMNS,  
NON EMPTY {Hierarchize({[DataEmissione].[Anno].Members,  
[DataEmissione].[Semestre].Members, [DataEmissione].[Trimestre].Members,  
[DataEmissione].[Mese].Members})} ON ROWS  
FROM [Venduto]
```

Listato 9-8 - Tempi attesa medi di saldo delle fatture.

Come di evince dal Listato 9-8, nella clausola SELECT è stata specificata la misura [Measures].[GiorniPagamento] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) viene visualizzato l'anno, il semestre, il trimestre e il mese.

In Figura 9-8 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Anno	Semestre	Trimestre	Mese	GiorniPagamento
2011				101
	1			71
		1		48
			FEB	76
			MAR	32
		2		80
			MAG	80
	2			133
		3		153
			LUG	305
			AGO	30
			SET	193
		4		108
			NOV	108

Figura 9-8 - Tempi attesa medi di saldo delle fatture.

### Marginalità delle vendite rispetto ai costi dei prodotti.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-9.

```
SELECT  
NON EMPTY {Hierarchize({[Measures].[MarginalitaProdotto]})} ON COLUMNS,  
NON EMPTY {Hierarchize({[Prodotti.ProdottiGerarchia].[Categoria].Members},  
{[Prodotti.ProdottiGerarchia].[Prodotto].Members})} ON ROWS  
FROM [Venduto]
```

Listato 9-9 - Marginalità delle vendite rispetto ai costi dei prodotti.

Come di evince dal Listato 9-9, nella clausola SELECT è stata specificata la misura [Measures].[MarginalitaProdotto] da visualizzare sull'asse verticale (ON COLUMNS), mentre sull'asse orizzontale (ON ROWS) vengono visualizzati i prodotti divisi in categorie.

In Figura 9-9 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

Categoria	Prodotto	MarginalitaProdotto
Alcoholic		€ 1,07
	Ale Beer	€ ,68
	Cerveza Ale 0,5L	€ ,04
	Cerveza Lager 0,5L	€ 1,23
	Lager Beer	€ ,88
	Red Wine	€ 1,73
	Rose wine	€ 1,88
	Vino Blanco 0,75L	€ ,93
	Vino Rosado 0,75L	€ ,95
	Vino Tinto 0,75L	€ ,85
	White wine	€ 1,78
Bio		€ ,33
	Orange Juice bio	€ ,45
	Zumo de Naranja Bio 0,33L	€ ,28
Fruit juice		€ ,31

Figura 9-9 - Marginalità delle vendite rispetto ai costi dei prodotti.



## 9.3 Analisi Campagne promozionali

### Ricavato da campagne promozionali per sede “Capogruppo”.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-10.

```
SELECT
NON EMPTY {Hierarchize({[Measures].[ImportoNetto]})} ON COLUMNS,
NON EMPTY Hierarchize(Union(CrossJoin([CampagnaPromozionale].[Campagna
promozionale].Members, [Succursale.SuccursaleGerarchia].[Succursale].Members),
CrossJoin([CampagnaPromozionale].[Campagna promozionale].Members,
[Succursale.SuccursaleGerarchia].[Capogruppo].Members))) ON ROWS
FROM [Venduto]
```

Listato 9-10 - Ricavato da campagne promozionali per sede “Capogruppo”.

Come di evince dal Listato 9-10, nella clausola SELECT è stata specificata la misura [Measures].[ImportoNetto] da visualizzare sull’asse verticale (ON COLUMNS), mentre sull’asse orizzontale (ON ROWS), utilizzando la funzione **Union()**<sup>13</sup> con input la **CrossJoin()** con una generica campagna promozionale e una succursale e una **CrossJoin()** con una generica sede “capogruppo” e “Succursale”, vengono visualizzate le Campagne promozionale, le sedi “capogruppo e le sedi “succursale”.

In Figura 9-10 è presentato un esempio di report generato per rispondere a tale

Campagna promozionale	Capogruppo	Succursale	ImportoNetto
Nessuna Camapgna	F&B International Group		€ 34.183.695,38
		F&B España - Región Norte	€ 25.271.928,00
		F&B España - Región Sur	€ 4.254.339,00
		F&B US West Coast	€ 1.523.684,18
		F&B US East Coast	€ 3.133.744,20
Prendi2paghi1	F&B International Group		€ 721.410,50
		F&B España - Región Norte	€ 542.766,00
		F&B España - Región Sur	€ 100.155,00
		F&B US West Coast	€ 9.961,00
		F&B US East Coast	€ 68.528,50
Prendi3paghi2	F&B International Group		€ 308.322,00
		F&B España - Región Norte	€ 230.155,00
		F&B US West Coast	€ 57.346,50
		F&B US East Coast	€ 20.820.50

<sup>13</sup> Rest duplica

Figura 9-10- Ricavato da campagne promozionali per sede “Capogruppo”.

requisito di analisi.

### Totale dei nuovi clienti derivati da campagne per sede “Capogruppo”.

Considerato il requisito di analisi richiesto, la formulazione della query MDX per la sua esplicitazione è riportata nel Listato 9-11.

```
SELECT
NON EMPTY {Hierarchize({[Measures].[NuoviClienti]})} ON COLUMNS,
NON EMPTY Hierarchize(Union(CrossJoin([CampagnaPromozionale].[Campagna
promozionale].Members, [Succursale.SuccursaleGerarchia].[Capogruppo].Members),
CrossJoin([CampagnaPromozionale].[Campagna promozionale].Members,
[Succursale.SuccursaleGerarchia].[Succursale].Members))) ON ROWS
FROM [Venduto]
```

Listato 9-11 - Totale dei nuovi clienti derivati da campagne per sede “Capogruppo”.

Campagna promozionale	Capogruppo	Succursale	Nuovi Cienti
Prendi2paghi1	F&B International Group		2
		F&B España - Región Norte	0
		F&B España - Región Sur	1
		F&B US West Coast	0
		F&B US East Coast	1
Prendi3paghi2	F&B International Group		0
		F&B España - Región Norte	0
		F&B US West Coast	0
		F&B US East Coast	0
Sconto IVA	F&B International Group		2

Figura 9-11 - Totale dei nuovi clienti derivati da campagne per sede “Capogruppo”.

Come di evince dal Listato 9-11, nella clausola SELECT è stata specificata la misura [Measures].[NuoviClienti] da visualizzare sull’asse verticale (ON COLUMNS), mentre sull’asse orizzontale (ON ROWS) utilizzando la funzione **Union()** con input la **CrossJoin()** con una generica campagna promozionale e una succursale e una **CrossJoin()** con una generica sede “capogruppo” e “Succursale”, vengono visualizzate le Campagne promozionale, le sedi “capogruppo e le sedi “succursale”.

In Figura 9-11 è presentato un esempio di report generato per rispondere a tale requisito di analisi.

#### **Analisi incremento vendite per prodotto e per cliente.**

Considerato il requisito di analisi richiesto e dopo una consultazione con il committente, si è preferito formulare due query MDX, uno per l'analisi sui prodotti (Listato 9-12) e l'altro per l'analisi per cliente (Listato 9-13).

```
SELECT  
NON EMPTY {Hierarchize({[[CampagnaPromozionale].[Prendi2paghi1],  
[CampagnaPromozionale].[Prendi3paghi2], [CampagnaPromozionale].[Sconto IVA],  
[CampagnaPromozionale].[Sconto10], [CampagnaPromozionale].[Sconto15],  
[CampagnaPromozionale].[Sconto5], [CampagnaPromozionale].[Nessuna  
Camapgna]]})} ON COLUMNS,  
NON EMPTY CrossJoin([Prodotti.ProdottiGerarchia].[Prodotto].Members,  
{[Measures].[ImportoNetto]}) ON ROWS  
FROM [Venduto]
```

**Listato 9-12** - Analisi incremento vendite per prodotto.

```

SELECT
NON EMPTY {Hierarchize({[CampagnaPromozionale].[Prendi2paghi1],
[CampagnaPromozionale].[Prendi3paghi2], [CampagnaPromozionale].[Sconto IVA],
[CampagnaPromozionale].[Sconto10], [CampagnaPromozionale].[Sconto15],
[CampagnaPromozionale].[Sconto5], [CampagnaPromozionale].[Nessuna Camapgna]}))} ON
COLUMNS,
NON EMPTY CrossJoin([Prodotti.ProdottiGerarchia].[Prodotto].Members,
{[Measures].[ImportoNetto]}) ON ROWS
FROM [Venduto]

```

**Listato 9-13** - Analisi incremento vendite per cliente.

Nelle Figura 9-12 e Figura 9-13 vengono presentati degli esempi di report generati per rispondere a tali requisiti di analisi.

Prodotto	MeasuresLevel	Nessuna Camapgna	Prendi2paghi1	Prendi3paghi2	Sconto IVA	Sconto10
Ale Beer	ImportoNetto	€ 291.720,00		€ 4.080,00		
Cerveza Ale 0,5L	ImportoNetto	€ 2.739.720,00	€ 71.400,00	€ 67.320,00	€ 10.200,00	
Cerveza Lager 0,5L	ImportoNetto	€ 1.879.680,00	€ 42.240,00			
Lager Beer	ImportoNetto	€ 587.479,20	€ 47.520,00			
Red Wine	ImportoNetto	€ 1.274.280,00		€ 51.800,00	€ 9.324,00	
Rose wine	ImportoNetto	€ 53.485,00		€ 5.630,00		
Vino Blanco 0,75L	ImportoNetto	€ 3.885.570,00	€ 53.300,00	€ 26.650,00	€ 13.325,00	
Vino Rosado 0,75L	ImportoNetto	€ 2.646.100,00				€ 140.750,00
Vino Tinto 0,75L	ImportoNetto	€ 4.369.330,00	€ 139.860,00	€ 77.700,00	€ 12.950,00	
White wine	ImportoNetto	€ 436.527,00	€ 1.066,00	€ 15.190,50	€ 4.530,50	
Orange Juice bio	ImportoNetto	€ 250.580,00	€ 10.720,00			€ 670,00
Zumo de Naranja Bio 0,33L	ImportoNetto	€ 2.925.220,00	€ 52.260,00	€ 52.260,00	€ 20.100,00	
Pear Juice	ImportoNetto	€ 162.211,50	€ 5.085,00	€ 1.243,00		€ 1.356,00
Pineapple Juice	ImportoNetto	€ 273.698,00			€ 2.051,00	€ 2.362,00

**Figura 9-12** - Analisi incremento vendite per prodotto.

Cliente	MeasuresLevel	Nessuna Camapgna	Prendi2paghi1	Prendi3paghi2	Sconto IVA	Sconto10	Scor
Alimentos y Supermercados, S.A	ImportoNetto	€ 8.574.648,00	€ 30.976,00	€ 177.895,00	€ 43.040,00	€ 160.905,00	€ 32.5
Restaurantes Luna Llena, S.A.	ImportoNetto	€ 4.254.339,00	€ 100.155,00		€ 29.880,00	€ 1.490,00	€ 54.1
Hoteles Buenas Noches, S.A.	ImportoNetto	€ 16.697.280,00	€ 511.790,00	€ 52.260,00	€ 69.395,00	€ 87.960,00	€ 59.7
Sleep Well Hotels, Co.	ImportoNetto	€ 1.782.403,70	€ 31.903,50	€ 5.630,00	€ 14.448,10	€ 21.239,60	
Healthy Food Supermarkets, Co.	ImportoNetto	€ 1.351.340,50	€ 36.625,00	€ 15.190,50	€ 3.051,00	€ 12.694,00	
More Food Restaurants, Co.	ImportoNetto	€ 1.523.881,40	€ 30.001,00	€ 57.340,00	€ 3.444,00	€ 2.788,00	

Figura 9-13 - Analisi incremento vendite per cliente.

## 9.4 Report

Non tutte le esigenze del cliente possono essere risolte utilizzando un report semplice. Come accade nella maggior parte dei casi, bisogna creare report complicati e di difficile gestione per il personale dell'azienda committente che non conosce la tecnologia usata.

Per questo motivo si è scelto di creare dei report ad hoc statici destinati alla sola consultazione. I parametri contenuti in essi sono stati decisi con il committente a consuntivo.

**Report vendite per: prodotto, quantità, importo, suddiviso per periodo.**

Colonne	Anno	Semestre	Trimestre	Mese	Importo Netto	Quantita
Filtri	Categoria	Prodotto				

**Figura 9-14 - Report vendite per: prodotto, quantità, importo, suddiviso per periodo.**

109

Nella Figura 9-15 viene presentato un esempio di report generato.

**Figura 9-15 - Report vendite per: cliente, quantità, importo, suddiviso per periodo.**

110

Nella Figura 9-15 viene presentato un esempio di report generato.

			1								
							1				
									FEB		
AreaGeografica	Citta	Agente	ImportoNetto	Quantita	ImportoNetto	Quantita	ImportoNetto	Quantita	ImportoNetto	Quantita	In
Spain			€ 21.895.784,10	11.432.350	€ 16.177.095,40	9.179.960	€ 4.323.319,40	1.428.760	€ 3.465.440,00	973.000	€
	Barcelona		€ 4.821.021,00	2.971.700	€ 2.509.521,00	1.829.700	€ 99.671,00	18.700			€
		Giuseppe López	€ 4.821.021,00	2.971.700	€ 2.509.521,00	1.829.700	€ 99.671,00	18.700			€
	Pamplona		€ 12.084.603,40	6.316.160	€ 11.221.353,40	5.791.160	€ 3.536.086,40	999.760	€ 3.465.440,00	973.000	€
		Andrea Romeo	€ 6.596.842,40	4.288.960	€ 6.596.842,40	4.288.960	€ 2.235.886,40	444.760	€ 2.165.240,00	418.000	€
		Antonio Rossi	€ 3.073.345,00	956.500	€ 2.558.695,00	851.500	€ 561.000,00	275.000	€ 561.000,00	275.000	
		Federica Giusti	€ 2.414.416,00	1.070.700	€ 2.065.816,00	650.700	€ 739.200,00	280.000	€ 739.200,00	280.000	
	San Sebastián		€ 2.406.796,70	775.590	€ 878.412,00	444.800	€ 302.002,00	158.300			€
Leandro		€ 1.480.993,50	288.150								

**Figura 9-16 - Report vendite per: agente, quantità, importo, suddiviso per periodo.**



## 10 CONCLUSIONI

Questo lavoro di tesi, svolto in ambito aziendale, ha avuto come dominio di applicazione le attività di un'impresa della distribuzione alimentare.

L'obiettivo principale era quello di progettare e costruire un sistema di Business Intelligence utilizzando solo prodotti *open source*. Esso doveva permettere la gestione dei dati generati dagli ordini di vendita, dal venduto e dalle campagne promozionali al fine di supportare le decisioni strategiche ed operative dei responsabili di questi settori.

Grazie agli strumenti messi a disposizione da questo sistema di BI, l'azienda è in grado di intervenire tempestivamente e con cognizione di causa al presentarsi di determinati eventi. Grazie alla BI le informazioni presenti nel sistema informativo potranno essere maggiormente sfruttate portando ad un miglioramento della capacità decisionale aziendale.

Il focus principale del lavoro di tesi è stato il processo di creazione del *data warehouse* aziendale. Questo ha permesso di definire una solida base dalla quale partire per poter analizzare i dati.

La progettazione del *data warehouse* è stata realizzata focalizzandosi le richieste degli utenti, ma tenendo presenti anche eventuali sviluppi futuri. Questo ha permesso di realizzare una struttura dati adatta ad una vasta gamma di analisi per monitorare e valutare le *performance* aziendali.

Per concludere, l'esperienza lavorativa è stata particolarmente interessante e gratificante, sia perché mi ha concesso la possibilità di conoscere più da vicino la complessa situazione aziendale, comprendendone i complicati meccanismi, sia

perché ho potuto migliorare la conoscenza in materia di progettazione e sviluppo di un *data warehouse*.

### **Sviluppi futuri**

Il lavoro svolto è la base per creare cruscotti grafici *on line*, raggiungibili dai responsabili aziendali grazie ad un *web browser*.

La struttura del *data warehouse*, permetterà, quando si renderanno disponibili una quantità maggiore di dati, riferita a più anni, di ampliare e completare il sistema di reporting in modo da permettere anche analisi di confronto tra più esercizi.

## I. BIBLIOGRAFIA

- [Alb12] A. Albano. Basi di dati di supporto alle decisioni. Università di Pisa, 2012.
- [Bou09] R. Bouman e J. Dongen. Pentaho Solution. Wiley, 2009.
- [Kim02] R. Kimball e M. Ross. The Data warehouse Toolkit. Wiley, 2002.
- [Msd12] *Sito ufficiale Microsoft MSDN*, 2012.
- [Rug12] S. Ruggieri. Analisi dei processi aziendali. Università di Pisa, 2012.

## II. RINGRAZIAMENTI

Le persone che sento di dover ringraziare per il loro sostegno sono veramente tante. Non potrei che cominciare col ringraziare mamma e papà che mi hanno dato la fiducia necessaria per intraprendere il mio percorso di studi, e per avermi sostenuto moralmente ed economicamente in tutti questi anni.

[maco da qualche parete]

Un ringraziamento amoroso va a Gabriella, mia compagna di vita fin dai primi anni di Università, che ha supportato e sopportato (soprattutto quando c'era da correggere le mie relazioni) e dato tante dritte per superare gli esami.

Un ringraziamento va all'associazione AEGEE, che mi ha fatto conoscere tantissime persone e continua a farlo tutt'ora.

Vorrei ringraziare anche al gruppo degli appassionati di giochi da tavolo, con i quali superavo lo stress giornaliero.

Infine, un ringraziamento va anche a Stefano che mi ha fatto toccare con mano il mondo del lavoro in questo campo e avuto molta pazienza nel spiegarmi alcuni concetti che non mi entravano in testa.

Un ultimo ringraziamento, ma solo in ordine di scrittura, va ai professori Salvatore Ruggieri e Antonio Albano per il prezioso contributo nella realizzazione di questo lavoro.

Concludo questa lista, ringraziando tutti gli amici che ho incontrato in questi lunghi anni di università, e sapendo di aver dimenticato qualcuno, rivolgo ancora un sentito grazie a tutti.