Sommario

Premessa

La necessità di conservare dati e informazioni in modo permanente, perché potranno essere utili in momenti successivi, è un problema molto evidente nel mondo moderno e riguarda ormai un numero elevatissimo di persone ed agenti economici: infatti questo vale sia per dati di tipo personale, o documenti utili per la vita di un ente o una qualunque impresa.

In un'azienda l'esecuzione delle normali attività sia amministrative che operative, la definizione e la scelta delle politiche commerciali, di quelle finanziarie e di quelle relative al personale, sono strettamente legate all'elaborazione di insiemi di dati che vengono chiamati **archivi**.

La conservazione e il successivo utilizzo di dati può costituire una fonte preziosa per prendere decisioni o per aumentare l'attività commerciale.

Lo step successivo alla conservazione dei dati è l'utilizzo in maniera efficace ed efficiente degli stessi. La parola d'ordine in questo settore oggi giorno è la Business Intelligence. Per essere implementato un sistema di BI deve avere a disposizione dei dati da utilizzare, e questi sono popolati in un *magazzino* detto Data Warehouse. Quindi alla base di una soluzione BI vi è proprio l'implementazione di un Data Warehouse o Data Mart.

La BI apporta un notevole valore economico alle aziende, le aiuta nella pratica acquisire una migliore comprensione di se stessa. Più in generale, si riferisce alla competenza, tecnologia, applicazioni, coinvolte a portare alla luce tale *comprensione*.

Definizione del ricercatore Hans Peter (IBM):

la capacità di cogliere le interrelazioni dei fatti esposti in modo da orientare l'azione verso un obiettivo desiderato. La Business Intelligence non è solo concentrata sotto il profilo tecnologico, è importante nel capire le relazioni tra i diversi aspetti dell'azienda in modo che si può guidarla verso obiettivi specifici, come ad esempio l'incremento della quota di mercato e il miglioramento della customer satisfaction. Quello che è di assoluta importanza è che la BI è di supporto al processo decisionale in un'azienda.

L'avvento di questo tipo di soluzione è dovuto al fatto che le aziende stanno annegando nei dati che registrano nei "*silos*": i dati del libro paga, dati finanziari, i dati dei clienti, i dati del venditore, e così via. Queste banche dati sono tipicamente sintonizzate per le singole operazioni, come

ad esempio il recupero di un unico ordine del cliente, o per le operazioni batch specifiche, quali l'elaborazione di buste paga, alla fine di ogni mese. Questi database non sono progettati per comunicare l'uno con l'altro, per consentire agli utenti di esplorare i dati in modo inusuale, o per fornire una sintesi di alto livello dei dati in un istante.

La BI tira fuori tutti i dati insieme e li mette in relazione. I dati possono sembrare non collegati, ma in realtà tutta l'attività in qualche modo è quasi sempre legata. Non si generano nuovi dati, si rende semplicemente più facile esplorare le relazioni tra i dati che sarebbero trascurate dal *decision maker*.

L'utilizzo di questa soluzione può alimentare direttamente nella pianificazione di sistemi decisionali, contribuendo a definire budget, obiettivi di vendita, ecc. Se adeguatamente progettata e attuata rapidamente può fornire un elevato ritorno sugli investimenti (ROI). Inoltre la cosa importante è anche il fatto che non bisogna distruggere i vecchi sistemi a supporto delle decisioni presenti in azienda.

Capita l'importanza della BI, bisogna vedere ora come poterla implementare. Alla base vi è la conservazione dei dati in maniera efficace ed efficiente in un magazzino dati. Senza questa tecnologia non sarebbe possibile pensare ad un sistema di BI e a tutti i vantaggi da essa derivanti.

Nei seguenti capitoli mi occuperò più dettagliatamente di analizzare, prima dal punto di vista teorico e poi pratico, come si progetta e gestisce un Data Warehouse. Infatti, successivamente sarà creato un Data Warehouse con successivo popolamento dello stesso con dati relativi alle vendite di un'azienda di distribuzione di prodotti farmaceutici (CEDIFARME). La popolazione del Data Warehouse avverrà tramite i dati presenti sui database di un sistema operazionale già presente in azienda. Essendo dati relativi a dimensioni temporali che vanno nell'ordine di un decennio, i tempi di risposta di una qualsiasi interrogazione sono pressoché giornalieri. Con la creazione di un Data Warehouse, si vuole superare l'inefficienza dovuta ai tempi di risposta, e si vuole dare al management un efficace strumento per effettuare interrogazioni utili e che siano a supporto delle decisioni.

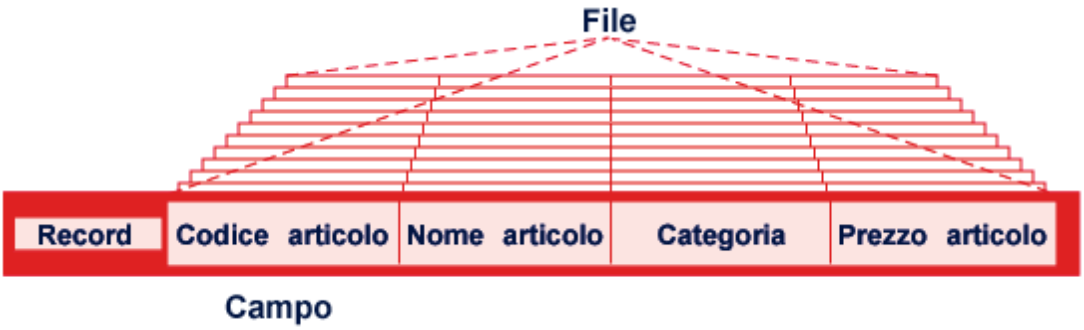
Capitolo 1

Verso il Data Warehouse

1. Nozioni di base

In un archivio le informazioni, in genere, sono raggruppate secondo un'unità logica. L'insieme di informazioni logicamente organizzate e riferibili ad un unico soggetto vengono chiamate **record**; mentre le singole informazioni che compongono il record si chiamano **campi**. Quindi riepilogando, abbiamo i campi, che non sono altro che la più piccola classe di dati (ad es. il "Cognome" di una persona); il record invece l'insieme di campi aventi un legame logico (ad es. una "scheda anagrafica"). L'insieme omogeneo di record dello stesso tipo (ad es. un "archivio anagrafico") costituiranno un **file**.

Figura 1- Esempio di File



I *limiti* di questa organizzazione dei dati sono derivanti da:

- **Ridondanza** = ossia gli stessi dati compaiono in maniera duplicata
- **Incongruenza** = nel caso in cui un dato venga aggiornato in un archivio e non in un altro, oppure siano presenti valori diversi per lo stesso dato
- **Inconsistenza** = cioè in dati a disposizione non sono più affidabili, perché non si sa in modo certo quale dei diversi valori sia quello corretto.

2. I DATABASE

I **Database** sono gli archivi di dati, organizzati in modo integrato attraverso tecniche di modellazione dei dati e gestiti sulle memorie di massa attraverso appositi *software*, con l'obiettivo di raggiungere una *grande efficienza* nel trattamento e ritrovamento dei dati. Quindi mentre i file sono

ancorati al supporto fisico, il database è indipendente dalla locazione e dalla struttura fisica dei dati stessi.

A grandi linee possiamo dire che il database è una collezione di archivi di dati ben organizzati e ben strutturati, in modo che possano costituire una base di lavoro per utenti diversi con programmi diversi.

Quando si parla di *efficienza* di un database significa garantire:

- **Consistenza** = i dati contenuti negli archivi devono essere significativi ed essere effettivamente utilizzabili nelle applicazioni dell'utente.
- **Sicurezza** = impedire che i dati del database vengano danneggiati da interventi accidentali e non autorizzati.
- **Integrità** = garantire che le operazioni effettuate sul database da utenti autorizzati non provochino una perdita di consistenza dei dati.

In precedenza è stato accennato che a gestire i database sono appunto dei software. Questi sono i cosiddetti **DBMS**, ossia Data Base Management System. Consentono all'utente di collocarsi in una posizione più lontana dall'hardware, dalle memorie di massa e dal sistema operativo e più vicina all'applicazione che usa i dati contenuti negli archivi.

La gestione degli archivi così presentata ha tali caratteristiche fondamentali:

- Indipendenza dalla struttura fisica dei dati
- Indipendenza dalla struttura logica dei dati
- Utilizzo da parte di più utenti
- Eliminazione della ridondanza
- Eliminazione della inconsistenza
- Facilità di accesso
- Integrità dei dati
- Sicurezza dei dati
- Uso di linguaggi per la gestione del database (attraverso comandi per la manipolazione e interrogazione dei dati per ottenere informazioni desiderate).

I DBMS presentano sistemi software che gestiscono i dati di un sistema informativo, assumendo il ruolo di interfaccia verso i programmi utente.

Le componenti sono:

– *linguaggio per la definizione o descrizione dei dati* (**DDL**, Data Definition Language)

– *linguaggio per la manipolazione dei dati* (**DML**, Data Manipulation Language)

– *linguaggio di descrizione delle modalità di memorizzazione* (**DSDL**, Data Storage Description Language).

Gli utenti interagiscono con un database per mezzo del DML (Data Manipulation Language), mentre il gestore del database (DBA, Data Base Administrator) dispone di strumenti per descrivere il modo in cui i dati sono logicamente organizzati e fisicamente memorizzati nel database (DDL, Data Definition Language, e DSDL, Data Storage Description Language).

3. MODELLI PER IL DATABASE

La soluzione più semplice è quella di costruire un database con una struttura di dati formata da un unico file. Questa struttura è detta **flat file**, è adatta solo per le basi di dati estremamente semplici. È questo il caso di un foglio elettronico nelle versioni moderne.

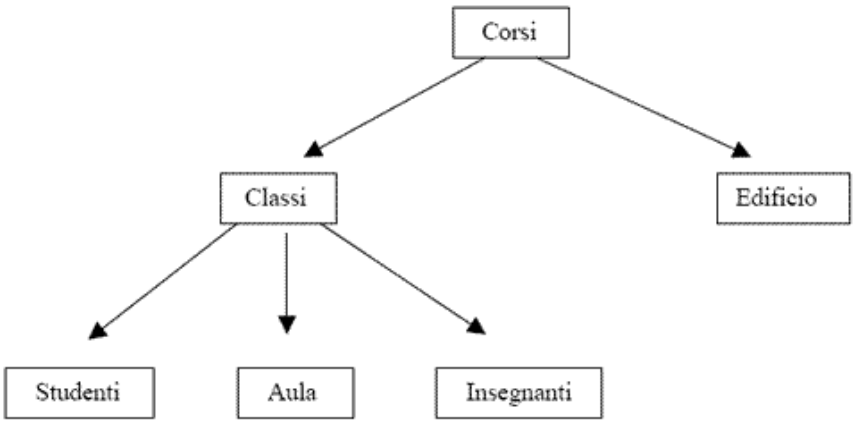
Dal 1960 in poi sono emersi principalmente quattro tipi di modelli per le basi di dati:

- Modello gerarchico
- Modello reticolare
- Modello relazionale
- Modello a oggetti

3.1 Modello gerarchico

È adatto per rappresentare situazioni nelle quali è possibile fornire all'insieme dei dati una struttura nella quale ci sono entità che stanno in alto ed entità che stanno in basso, secondo uno schema ad albero, nel quale i nodi rappresentano le entità e gli archi rappresentano le associazioni. Nella pratica l'entità è un file, l'istanza è un record e gli attributi sono i campi del record.

Figura 2 - Esempio Modello Gerrchico



Il modello gerarchico è particolarmente adatto a rappresentare le associazioni 1:N (uno a molti).

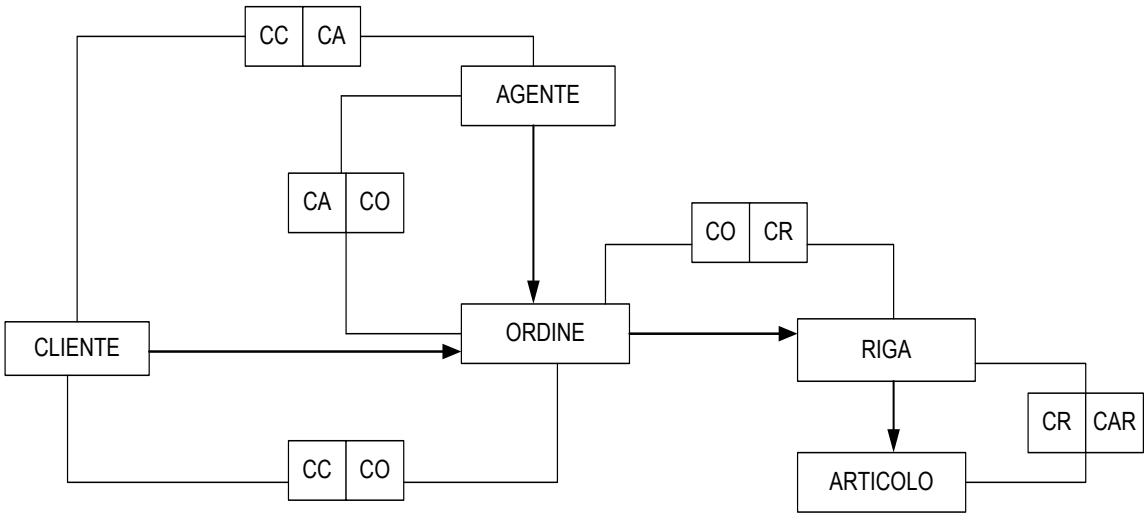
Presenta dei limiti, soprattutto nella rigidità della struttura di dati creata, che talvolta non riesce ad evitare la ridondanza dei dati.

3. 2 Modello reticolare

Le entità rappresentano i nodi e le associazioni rappresentano gli archi di uno schema a grafo orientato: cioè un'estensione del modello di albero gerarchico, essendo consentite anche le associazioni tra entità che stanno in basso, e non solo dall'alto verso il basso. La differenza principale rispetto al modello gerarchico è nel fatto che un record figlio può avere un numero qualsiasi di padri: in questo modo vengono evitate situazioni di ripetizione di dati uguali.

[1]

Figura 3 - Esempio Modello Reticolare



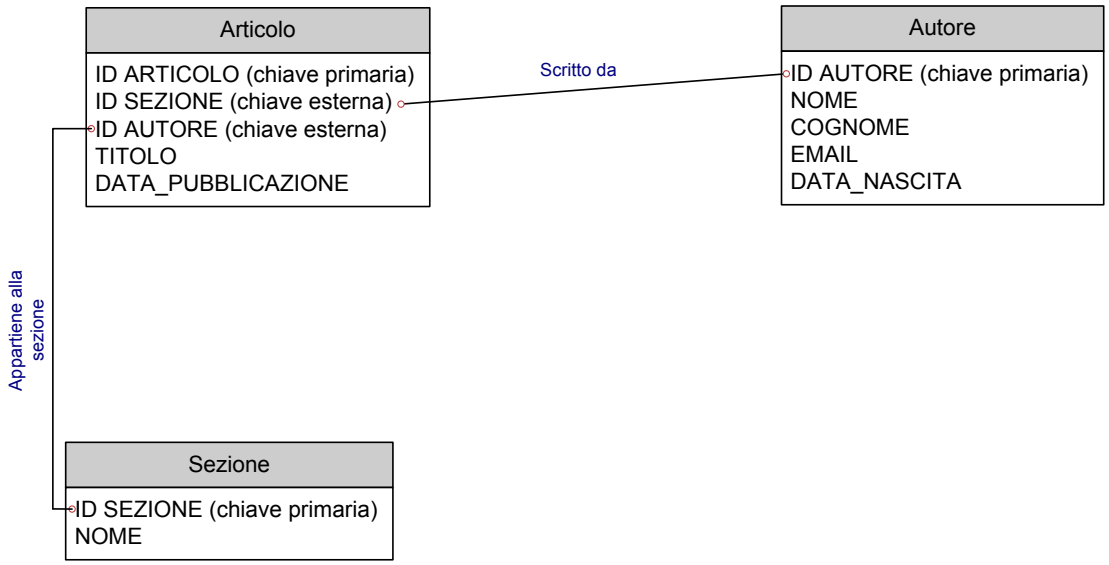
È una soluzione molto complessa, risulta pertanto più difficile l'implementazione e la costruzione del software applicativo.

[1] *CC = codice cliente; CA = codice agente; CO = codice ordine; CR = codice riga; CAR = codice articolo*.

3.3 Modello relazionale

Rappresenta il database come un insieme di tabelle. È il modo più semplice ed efficace.

Figura 4 - Esempio Modello Relazionale



Il modello relazionale si bassa su alcuni concetti fondamentali tipicamente matematici e assegna grande importanza all'uso rigoroso del linguaggio matematico, con due obiettivi importanti:

- Utilizzare un linguaggio conosciuto a livello universale, quale è il linguaggio matematico
- Eliminare i problemi di ambiguità nella terminologia e nella simbologia

Le operazioni sui database gerarchici e reticolari sono complesse e agiscono sui singoli record e non su gruppi di record. I modelli non relazionali sono basati sulla programmazione di applicazioni, l'utente deve specificare i percorsi per ritrovare i dati e la velocità nel ritrovare le informazioni dipende dal software di gestione; nel modello relazionale, invece, i percorsi per le interrogazioni sono a carico del sistema.

3.4 Modelli di database orientati ad oggetti

Sono database che gestiscono **oggetti** e **classi**.

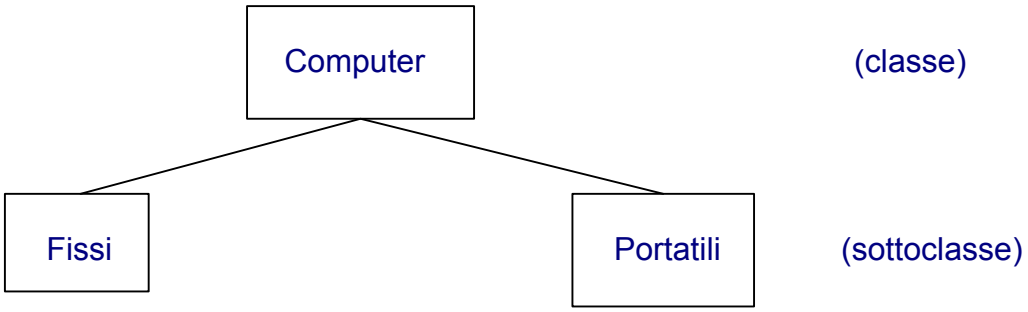
Oggetto = Un oggetto è **autoconsistente** se possiede all'interno tutto quello che serve al suo funzionamento. Entità che incorpora al suo interno 2 caratteristiche:

- *Proprietà o attributi informativi* (caratteristiche ben precise e ben catalogabili)
- *Metodi o azioni* (un oggetto viene visto perché quello che deve fare lo sa fare, quindi sono le azioni che l'oggetto deve saper svolgere. Le azioni possono avere come input attributi provenienti dall'esterno).

Classe di oggetti = all'interno ad es. della classe computer vi è l'oggetto preciso e concreto. L'oggetto deriva da una classe.

Sottoclassi = non sono ancora oggetti.

Figura 5 - Database orientato ad oggetti



Se devo definire delle proprietà o metodi per una classe, non devo quindi ridefinirle per tutti gli oggetti nella sottoclasse e così via. Quanto definito in alto non lo si fa anche in basso. Un oggetto deriva le proprietà e metodi di tutte di tutte le proprietà e metodi della classe che lo precede.

Questo approccio fa definire i sistemi informativi molto complessi. L'oggetto è caratterizzato dall'**incapsulamento** cioè nascondere l'informazione. Nel senso che come l'automobile accelera non interessa a nessuno.

Tipologie di creazione:

Top-down = costruisco il database partendo dall'alto verso il basso

Buttom-up = costruisco il sistema informativo mattoncino per mattoncino dal basso verso l'alto.

Esiste anche un approccio misto "*sandwitch*" che è una via di mezzo.

Una seconda caratterizzazione dell'oggetto è l'**ereditarietà** cioè che significa che un oggetto appartiene ad una classe, che poi appartiene ad un'altra classe e così via. Un oggetto fa sempre riferimento alla proprietà e metodo più vicina a lui. Se ci sono attributi e metodi ridefiniti con nomi uguali si nascondono quelli a livello superiore (es. accelera → suv accelera → berlina).

Una terza caratterizzazione dell'oggetto è il **polimorfismo**. Se vi è una orizzontalità nei metodi e attributi, cioè tutte le classi sono sullo stesso livello, allora l'oggetto utilizzerà le caratteristiche e metodi giusti in base alla tipologia dell'oggetto. Polimorfismo è la presenza contemporanea di metodi differenti, ma con lo stesso nome allo stesso livello gerarchico. In realtà non c'è ambiguità.

Il problema si ha quando ci sia ereditarietà generalizzata, cioè che una classe intermedia debba avere le caratteristiche e metodi di più classi che le stanno sopra. Si definiscono le caratteristiche e attributi solo in via dinamica (a runtime).

Capitolo 2

Il Data Warehouse

1. COS'È IL DATA WAREHOUSE

Quando siamo di fronte a grandi imprese che gestiscono grandi quantità di dati e soprattutto la velocità nel disporre di tali dati è essenziale per la competitività nel proprio settore di appartenenza, l'utilizzo di un database risulterebbe inconveniente, sia in termini di organizzazione dello stesso, sia in termini di tempi di risposta. Risulterebbe inefficiente perché ad esempio un'interrogazione su un database da parte di un soggetto decisore, produrrebbe un output anche a distanza di ore o giorni.

Infatti i dati gestiti e conservati all'interno dei sistemi transazionali non si prestano facilmente al tipo di analisi di cui gli utenti hanno bisogno. Le analisi che potrebbero essere condotte utilizzando i dati di tali sistemi sono dunque soggette a limitazioni causate dall'impostazione degli stessi.

Ed è qui che nasce l'esigenza per determinate imprese di implementare un sistema informativo aziendale che preveda l'utilizzo di un Data Warehouse.

Un **Data Warehouse** (dall'inglese *magazzino dati*), quindi, è un archivio informatico contenente i dati di un'organizzazione. Sono progettati per consentire di produrre facilmente relazioni ed analisi. L'approccio dei Data Warehouse si sta rivelando un'eccellente via per spostare il confine dell'elaborazione operazionale negli ambienti decisionali.

Un Data Warehouse risponde alle esigenze, in continuo aumento, di analizzare le informazioni relative all'andamento dell'azienda e di fare ciò facilmente, rapidamente e in maniera corretta. Il Data Warehouse consiste in una copia di dati provenienti dai sistemi transazionali, memorizzati in maniera tale da favorirne l'accesso a coloro (utenti e/o applicazioni) che devono prendere delle decisioni basate su di essi.

Un Data Warehouse ha ragione di esistenza per via dei limiti che presentano i sistemi operazionali, ossia:

- **Continua evoluzione dei dati** = i dati in un sistema transazionale non sono affidabili se usati all'interno di un processo decisionale; essi cambiano in continuazione per via delle operazioni effettuate attraverso l'OLTP.

- **Tempi lunghi di risposta delle interrogazioni =** in un sistema transazionale i dati sono distribuiti su numerose tabelle per permettere l'efficace aggiornamento. In un'interrogazione complessa i dati devono essere ricercati su più tabelle, con un conseguente rallentamento della ricerca.
- **Mancanza di dati storici adeguati** = le applicazioni di un sistema transazionale sono progettate per gestire processi riguardanti il momento attuale dell'azienda, senza fare riferimento a dati nel passato, che non vengono conservati.
- **Dati registrati in formati diversi** = applicazioni diverse possono usare tecnologie e piattaforme diverse, rendendone onerosa l'integrazione in un formato comune.

Le tecniche di datawarehousing sono scaturite dal bisogno di ovviare ai limiti dei sistemi transazionali.

Un Data Warehouse raccoglie i dati dai diversi sistemi transazionali, integra i dati in insiemi logici di pertinenza degli utenti finali, memorizza i dati in modo accessibile e di facile comprensione, fornisce accesso diretto ai dati da parte degli utenti attraverso potenti strumenti grafici di interrogazione e reportistica.

Le tecniche di datawarehousing permettono un accesso ai dati senza interferire con lo svolgimento delle operazioni da parte dei sistemi, le quali si rivelano molto spesso critiche per l'andamento dell'azienda.

Il risultato è la possibilità offerta all'utente finale di prendere delle decisioni di miglior qualità più rapidamente, più facilmente e con meno errori.

DEFINIZIONE DATA WAREHOUSE

Volendo dare una definizione precisa di Data Warehouse possiamo vederlo come una raccolta di dati **integrata**, **subject oriented**, **time variant** e **non-volatile** di supporto ai processi decisionali (*Inmon*).

Integrata: requisito fondamentale di un Data Warehouse è l'integrazione dei dati raccolti. Nel Data Warehouse confluiscono dati provenienti da più sistemi transazionali e da fonti esterne. L'obiettivo dell'integrazione può essere raggiunto percorrendo differenti strade: mediante l'utilizzo di metodi di codifica uniformi, mediante il perseguimento di una omogeneità semantica di tutte le variabili, mediante l'utilizzo delle stesse unità di misura;

Orientata al soggetto: il Data Warehouse è orientato a temi aziendali specifici piuttosto che alle applicazioni o alle funzioni. In un Data Warehouse i dati vengono archiviati in modo da essere facilmente letti o elaborati dagli utenti. L'obiettivo, quindi, non è più quello di minimizzare la ridondanza mediante la normalizzazione, ma quello di fornire dati organizzati in modo tale da favorire la produzione di informazioni. Si passa dalla progettazione per funzioni ad una modellazione dei dati che consenta una visione multidimensionale degli stessi;

Variabile nel tempo: i dati archiviati all'interno di un Data Warehouse coprono un orizzonte temporale molto più esteso rispetto a quelli archiviati in un sistema operativo. Nel Data Warehouse sono contenute una serie di informazioni relative alle aree di interesse che colgono la situazione relativa ad un determinato fenomeno in un determinato intervallo temporale piuttosto esteso. Ciò comporta che i dati contenuti in un Data Warehouse siano aggiornati fino ad una certa data che, nella maggior parte dei casi, è antecedente a quella in cui l'utente interroga il sistema. Ciò differisce da quanto si verifica in un sistema transazionale, nel quale i dati corrispondono sempre ad una situazione aggiornata, solitamente incapace di fornire un quadro storico del fenomeno analizzato;

Non volatile: tale caratteristica indica la non modificabilità dei dati contenuti nel Data Warehouse che consente accessi in sola lettura. Ciò comporta una semplicità di progettazione del database rispetto a quella di un'applicazione transazionale. In tale contesto non si considerano le possibili anomalie dovute agli aggiornamenti, né tanto meno si ricorre a strumenti complessi per gestire l'integrità referenziale o per bloccare record a cui possono accedere altri utenti in fase di aggiornamento.

2. Gestione dei dati attraverso il DATA WAREHOUSE

I flussi dei dati in un Data Warehouse sono:

- *Inflow* = flusso in entrata
- *Upflow* = flusso in entrata di dati aggregati o sommarizzati
- *Outflow* = flusso in uscita di dati verso utenti o applicazioni
- *Downflow* = flusso interno di ulteriore aggregazione dei dati
- *Down & Out* = flusso con cui i dati vengono rimossi

Il processo di flusso di dati per un Data Warehouse inizia con il trasferimento dei dati nello stesso. I dati sono raccolti all'interno dei sistemi transazionali e vengono immessi nel Data Warehouse. Questo processo è chiamato *Inflow*.

I dati in un sistema transazionale sono, di norma, a livello di dettaglio. Parte di essi viene aggregata e sommarizzata per offrire agli utenti un tempo di risposta più rapido. I dati così trasformati vengono trasferiti nel Data Warehouse, e questo processo è chiamato *Upflow*.

I dati memorizzati in un Data Warehouse vengono messi a disposizione degli utenti finali. Questi ultimi, con l'ausilio di strumenti di interrogazione e analisi, possono ricevere un flusso di dati in uscita dal Data Warehouse. Questo processo è chiamato *Outflow*.

I dati memorizzati in un Data Warehouse possono essere ri-memorizzati sullo stesso in un formato di ulteriore aggregazione prima di essere rimossi. Ciò accade quando i dati sono vecchi e non vengono più usati con sufficiente frequenza da giustificarne la loro presenza. Tale processo è detto *Downflow*.

I dati originali di un processo di Downflow vengono rimossi dal Data Warehouse e trasferiti su supporti magnetici esterni. I dati non vengono semplicemente cancellati e quindi irrimediabilmente persi: se così fosse si perderebbe la memoria storica dell'azienda contravvenendo ai principi stessi del datawarehousing. Questo è il processo chiamato *Down & Out*.

3. Tipi di applicazioni utilizzati da un DATA WAREHOUSE

Le applicazioni sono categorizzate in base ai requisiti commerciali che il sistema deve prendere in considerazione. Le applicazioni di datawarehousing sono raggruppate nelle categorie di:

- **Produttività individuale** = Le applicazioni usate per elaborare e presentare i dati sul PC di un utente sono normalmente sviluppate in un ambiente indipendente e accedono e manipolano dei volumi limitati di dati.
- **Interrogazioni sui dati e reportistica =** questo tipo di applicazioni è usato per interrogazioni di limitata complessità e per reports su dati storici o abbastanza recenti.
- **Pianificazione e analisi =** è usato per analisi complesse di dati storici e per funzioni di pianificazione e previsione nel futuro basate su quegli stessi dati storici. Il risultato è la pianificazione e la previsione di eventi futuri, simulazioni, valutazioni di processi e opportunità. Queste applicazioni sono conosciute come OLAP (On line analytical processing).

Queste applicazioni sono strumenti di front-end per accedere, estrarre e analizzare i dati contenuti in un Data Warehouse.

4. Le basi di dati in un DATA WAREHOUSE

Una base di dati è un insieme, una collezione di dati in qualche modo correlati. In un Data Warehouse devono contenere un volume considerevole di dati, sia storici che attuali. Bisogna quindi affidarsi ad una tecnologia possente che riesce a gestire volumi elevati, al tempo stesso però offra flessibilità di accesso e di estrazione dei dati.

Vi sono due tipi di basi di dati che possono essere utilizzate per contenere e gestire i dati in Data Warehouse:

- **Relazionali**
- **Multi-dimensionali**

4.1 Basi di dati relazionali

Le basi di dati vengono chiamate in questo modo quando i dati elementari e le relazioni che esistono tra di essi, vengono registrati in forma tabellare. È utilizzato sia per i sistemi transazionali che per un Data Warehouse, ma i dati sono ottimizzati in modo diverso a causa dei requisiti diversi che caratterizzano i due tipi di sistemi.

I sistemi transazionali sono costruiti per gestire le operazioni giornaliere, quindi la base di dati è ottimizzata per permettere un aggiornamento efficace di singoli records. I dati sono *normalizzati*, cioè i dati di una tabella sono smistati su tabelle più piccole per evitare una ridondanza che potrebbe essere pericolosa per l'integrità dei dati stessi e per la rapidità delle operazioni di aggiornamento.

Un Data Warehouse è costruito per permettere un accesso istantaneo ai dati e garantire la flessibilità di interrogazioni e analisi. Se i dati fossero normalizzati l'accesso e le interrogazioni presenterebbero delle difficoltà in quanto essi devono essere raccolti da svariate tabelle, con un conseguente rallentamento nei tempi di risposta. In un Data Warehouse, quindi, i dati devono essere de normalizzati cioè memorizzati in un minor numero di grandi tabelle, per migliorare le prestazioni degli strumenti di interrogazione.

4.2 Basi di dati multidimensionali

È progettata particolarmente per andare incontro alle esigenze dei managers, direttori e analisti che vogliono vedere i dati in una maniera particolare, fare un numero elevato di interrogazioni specialistiche, ed analizzare i risultati utilizzando delle tecniche speciali.

Una base di dati multi-dimensionale, rappresenta i dati in essa contenuti come fossero dimensioni e non tabelle. È più facile da usare e più veloce di una base di dati relazionale. Oltre a fornire una visione dei dati in più dimensioni, questo tipo di base dati supporta la registrazione dei dati in vari livelli di aggregazione, in quanto le dimensioni sono strutturate gerarchicamente, inoltre supporta la possibilità di effettuare il *drill-down* e il *roll up* dei dati.

Funzioni che permettono ad un utente di navigare fino ad un singolo elemento di una dimensione e di visualizzarlo sia a livello di dettaglio che di aggregazione.

***Drill – down** = dal livello di aggregazione ai dettagli che lo compongono*

***Roll – up** = il procedimento inverso.*

Le basi di dati multi-dimensionali sono particolarmente indicate quando esistono molte interrelazioni tra le dimensioni della base di dati. In caso contrario una base dati relazionale è più efficiente.

5. Obiettivi del DATA WAREHOUSE

L'obiettivo dell'approccio di datawarehousing, è a tutti gli effetti l'utente: il Data Warehouse si prefigge di permettere un accesso flessibile ai dati in un ambiente di business.

I dati in un Data Warehouse (come detto in precedenza) possono essere definiti:

- Mirati verso un soggetto definito (subject oriented)
- Integrati (integrated)
- Permanenti o anche statici (non volatile)
- Con una profondità temporale (time – variant)

I dati in un sistema transazionale sono mirati a un'applicazione e cioè mirano a fornire un supporto ad un processo applicativo. I dati in un Data Warehouse sono invece mirati ad un soggetto particolare, e cioè mirano a fornire supporto ad una decisione.

I dati in un Data Warehouse sono il risultato di un consolidamento di dati provenienti da diversi sistemi transazionali. Mentre un sistema transazionale sostituisce continuamente vecchi dati con dei nuovi, un Data Warehouse assorbe i nuovi dati e li integra con quelli già presenti. Infatti i dati contenuti in un Data Warehouse sono un misto tra dati correnti e dati storici.

6. I componenti del DATA WAREHOUSE

Un Data Warehouse non posside una struttura predefinita, ma essa è determinata dai componenti che vengono utilizzati. Si può decidere di integrare in un Data Warehouse solo quei componenti di cui si ha bisogno, secondo le necessità e di sviluppare solo quelli che non sono disponibili sul mercato.

Per implementare un Data Warehouse bisogna innanzitutto capire quali sono le necessità degli utenti. È necessario capire se e come i vari prodotti tecnologici esistenti sul mercato soddisfino o meno le necessità dei progettatori, degli sviluppatori, degli amministratori, e di tutti gli utenti dell'azienda ove il Data Warehouse sarà implementato.

Un Data Warehouse ha i seguenti componenti logici:

- Definizione dei dati (data definition)
- Acquisizione dei dati (data collection)
- Gestione dei dati (data management)
- Metadati
- Analisi

6.1 Data Definition

Questo componente è finalizzato alla progettazione e alla definizione dell'ambiente del Data Warehouse e quindi a progettare e definire la sua struttura, identificare le sorgenti di dati, definire le operazioni di pulizia dei dati e le regole di trasformazione che condizionano i dati in un formato fruibile dai processi decisionali.

Viene usato per progettare e definire la base di dati del Data Warehouse, e cioè per creare:

- Le entità (tabelle)
- Gli attributi (colonne)
- Gli identificativi (chiavi)

Questo componente è utilizzato inoltre per identificare le varie fonti dei dati sia all'interno dell'azienda che al suo esterno, e cioè:

- Dati contenuti nei sistemi transazionali
- Dati provenienti da operatori del settore

6.2 Data Collection

L'obiettivo di questo componente è l'acquisizione dei dati necessari al Data Warehouse e quindi la regolazione del flusso dei dati. I progettisti e gli sviluppatori del Data Warehouse usano questo componente per estrarre i dati dalle varie sorgenti, effettuare la pulizia e la trasformazione di questi ultimi e la loro mappatura sulle strutture create e caricarli nella base di dati.

L'estrazione di dati dalle sorgenti avviene attraverso l'analisi delle stesse utilizzando dei criteri di selezione appropriati applicati a dei programmi specifici o generici.

Dopo l'estrazione, questo componente si occupa della pulizia dei dati, e cioè:

- Rimuovere le incongruenze
- Aggiungere dati mancanti
- Assicurarsi che l'integrità dei dati sia mantenuta

Inoltre viene effettuata anche una trasformazione dei dati:

- Aggiunta di campi relativi al tempo (es. data di estrazione)
- Aggregazione di dati di dettaglio
- Derivazione di nuovi campi

La mappatura dei dati sulle strutture ed il loro caricamento nella base di dati sono anch'essi aspetti di questo componente. La mappatura può avvenire sia attraverso strumenti generici che attraverso programmi specializzati. Il caricamento può avvenire sia attraverso programmi di utilità specifici per la base di dati prescelta che attraverso programmi scritti ad hoc.

6.3 Data Management

Questo è un componente che fornisce servizi agli altri componenti, gestisce tutte le basi di dati all'interno del Data Warehouse.

I servizi offerti da tale componente sono:

- Derivazione di nuovi dati sommarizzati partendo da dati di dettaglio
- Distribuzione di dati verso le postazioni di lavoro degli utenti
- Applicazione di criteri di sicurezza
- Operazioni di ripristino in caso di perdita dei dati
- Archiviazione dei dati
- Controllo continuo sui dati

La gestione delle basi di dati si incarica di creare, accedere, estrarre, mantenere i dati presenti in tutto il Data Warehouse. Perché questi servizi siano efficaci, il DBMS deve essere in grado di processare grossi volumi di dati in modo efficiente e, in particolare, deve supportare accessi paralleli e sofisticati criteri di indicizzazione.

6.4 Metadati

Rappresentano tutte le informazioni che riguardano la massa di dati contenuti in un Data Warehouse. Possono essere paragonati al catalogo di una biblioteca, il quale aiuta il lettore a sapere se un libro esiste e su quale scaffale si trova. I metadati forniscono indicazioni sulla descrizione dei dati, sulla loro struttura e su dove essi sono registrati.

6.5 Analisi

Permette l'ottenimento dei benefici derivanti dall'implementazione di un Data Warehouse. Fornisce un supporto all'ottenimento di dati da parte dell'utente e alla loro analisi. Questo supporto consiste nel fornire accesso diretto ai dati nel Data Warehouse, visualizzare i dati attraverso viste multi-dimensionali e permettere interrogazioni ad hoc o predefinite.

Il componente analisi contiene strumenti di tipo OLAP finalizzati all'analisi dei dati contenuti nel Data Warehouse. Questi strumenti sono concepiti per velocizzare l'ottenimento, la sommarizzazione, l'analisi dei dati, e per presentare una vista multidimensionale, utilizzando un motore dello stesso tipo.

7. OLAP

Acronimo che sta per l'espressione *On-Line Analytical Processing*. Designa un insieme di tecniche software per l'analisi interattiva e veloce di grandi quantità di dati, che è possibile esaminare in modalità piuttosto complesse. Questa è la componente tecnologica base del Data Warehouse e, ad esempio, serve alle aziende per analizzare i risultati delle vendite, l'andamento dei costi di acquisto delle merci, al marketing per misurare il successo di una campagna pubblicitaria, ad una università i dati di un sondaggio ed altri casi simili. Gli strumenti OLAP si differenziano dagli OLTP per il fatto che i primi hanno come obiettivo la performance nella ricerca e il raggiungimento di un'ampiezza di interrogazione quanto più grande possibile; i secondi, invece, hanno come obiettivo la garanzia di integrità e sicurezza delle transazioni.

7.1 Funzionalità

La creazione di un sistema OLAP consiste nell'effettuare una fotografia di informazioni in un determinato momento e trasformare queste singole informazioni in dati multidimensionali.

Eseguendo successivamente delle interrogazioni sui dati così strutturati è possibile ottenere risposte in tempi decisamente ridotti rispetto alle stesse operazioni effettuate su altre tipologie di database, anche perché il DB di un sistema OLTP non è stato studiato per consentire analisi articolate.

Una struttura OLAP creata per questo scopo è chiamata "*cubo*" *multidimensionale*. Ci sono diversi modi per creare un cubo, ma il più conosciuto è quello che utilizza uno schema "a stella" dove al centro c'è la tabella dei "fatti" che elenca i principali elementi su cui sarà costruita l'interrogazione, e collegate a questa tabella ci sono varie tabelle delle "dimensioni" che specificano come saranno aggregati i dati.

Per esempio un archivio di clienti può essere raggruppato per città, provincia, regione; questi clienti possono essere relazionati con i prodotti ed ogni prodotto può essere raggruppato per categoria.

Il calcolo delle possibili combinazioni di queste aggregazioni forma una struttura OLAP che, potenzialmente, potrebbe contenere tutte le risposte per ogni singola combinazione. In realtà viene memorizzato solo un

numero predeterminato di combinazioni, mentre le rimanenti vengono ricalcolate solo al momento in cui quella richiesta viene materialmente effettuata.

Un sistema OLAP permette di:

- studiare una grande quantità di dati
- vedere i dati da prospettive diverse
- supportare i processi decisionali.

7.2 Tipi di sistemi OLAP

Esistono tre tipologie di sistemi OLAP: multidimensionale (MOLAP: Multidimensional OLAP), relazionale (ROLAP: Relational OLAP) e ibrido (HOLAP: Hybrid OLAP).

MOLAP è la tipologia più utilizzata e ci si riferisce ad essa comunemente con il termine OLAP. Utilizza un database di riepilogo avente un motore specifico per l'analisi multidimensionale e crea le "dimensioni" con un misto di dettaglio ed aggregazioni. Risulta la scelta migliore per quantità di dati ridotte, perché è veloce nel calcolare aggregazioni e restituire risultati, ma crea enormi quantità di dati intermedi.

ROLAP lavora direttamente con database relazionali; i dati e le tabelle delle dimensioni sono memorizzati come tabelle relazionali e nuove tabelle sono create per memorizzare le informazioni di aggregazione. È considerato più scalabile e richiede minor spazio disco e minore RAM, ma è lento nella fase di creazione tabelle e nel produrre il risultato delle interrogazioni.

HOLAP utilizza tabelle relazionali per memorizzare i dati e le tabelle multidimensionali per le aggregazioni "speculative". Si pone nel mezzo, è in grado di essere creato più velocemente di ROLAP ed è più scalabile di MOLAP.

La difficoltà nell'implementazione di un database OLAP parte dalle ipotesi delle possibili interrogazioni utente; scegliere la tipologia di OLAP, lo schema e creare una base dati completa e consistente è un'operazione complessa. Decisamente complicata per una base di utenza ampia ed eterogenea. Per venire incontro alle esigenze degli utenti, molti prodotti moderni forniscono una quantità enorme di schemi ed interrogazioni pre-impostate.

7.3 Caratteristiche dell'OLAP

Le funzioni di base di uno strumento OLAP sono:

- *Slicing*: è l'operazione di rotazione delle dimensioni di analisi. È un'operazione fondamentale per analizzare totali ottenuti in base a dimensioni diverse o se si vogliono analizzare aggregazioni trasversali;
- *Dicing*: è l'operazione di *estrazione* di un subset di informazioni dall'aggregato che si sta analizzando. L'operazione di dicing viene eseguita quando l'analisi viene focalizzata su una 'fetta del cubo' avente particolare interesse per l'analista. In alcuni casi l'operazione di dicing può essere 'fisica' nel senso che non consiste solo nel filtrare le informazioni di interesse ma anche nell'estrarle dall'aggregato generale per distribuirne i contenuti;
- *Drill-down*: è l'operazione di 'esplosione' del dato nelle sue determinanti. L'operazione di drill-down può essere eseguita seguendo due diversi percorsi: la *gerarchia* costruita sulla dimensione di analisi (p. es.: passaggio dalla famiglia di prodotti all'insieme dei prodotti che ne fanno parte) oppure la *relazione matematica* che lega un dato calcolato alle sue determinanti (p. es.: passaggio dal margine al ricavo e costo che lo generano). È comprensibile l'importanza di tale operazione ai fini analitici in termini di comprensione delle determinanti di un dato;
- *Drill-across*: è l'operazione mediante la quale si naviga attraverso uno stesso livello nell'ambito di una gerarchia. Come visto precedentemente, il passaggio dalla famiglia di prodotti alla lista dei prodotti è un'operazione di drill-down, il passaggio da una famiglia ad un'altra famiglia è un'operazione di drill-across;
- *Drill-through*: concettualmente simile al drill-down, è l'operazione mediante la quale si passa da un livello aggregato al livello di dettaglio appartenente alla base dati normalizzata. Molti venditori proclamano che i loro prodotti hanno la capacità, mediante l'operazione di drill-through, di passare dal Data Warehouse ai sistemi transazionali alimentanti. Tale operazione, anche se tecnicamente fattibile sotto una serie di condizioni abbastanza rilevanti, è poco sensata per le problematiche di security e di performance indotti nei sistemi transazionali stessi.

7.4 Punti deboli

I punti deboli degli strumenti OLAP sono:

- **Inaccessibilità/difficoltà ad accedere al livello atomico del dato:** gli strumenti OLAP funzionano molto bene su dati di sintesi, non è conveniente usarli su dati analitici;
- **Sistemi di backup / restore / security / rollback non molto sofisticati o inesistenti:** pur essendo in molti casi dei motori database, gli strumenti OLAP non hanno ancora raggiunto il livello di completezza dei database relazionali, principalmente perché, a differenza di questi ultimi, non hanno un paradigma concettuale di riferimento, ma sono soggetti alle interpretazioni dei diversi software vendor;
- **Richiede una struttura denormalizzata per funzionare in maniera efficiente:** i motori OLAP generano grandi masse di dati per il semplice fatto che per migliorare le prestazioni di accesso sono costretti a memorizzare chiavi ridondanti e sommarizzazioni;
- **Possibile proliferazione del codice SQL:** nel caso in cui il database su cui vengono effettuate le analisi OLAP non sia multidimensionale (MOLAP) ma sia relazionale (ROLAP), ognuna delle operazioni sopra descritte (slicing, dicing, drilling) provoca la generazione e l'esecuzione di query SQL estremamente complesse, che richiedono molte risorse di elaborazione.

8. I modelli e le strutture dei dati in un Data Warehouse

Un Data Warehouse di solito si basa su un modello di dati diverso da quello utilizzato dai sistemi transazionali. I dati includono:

- Le strutture tipiche delle basi di dati (tabelle, attributi e campi chiave)
- La rappresentazione delle relazioni esistenti tra queste diverse strutture

I modelli di dati normalmente utilizzati per la progettazione di una base di dati sono i modelli entità relazioni (ERM). Questo tipo di modello però presenta dei problemi: come nella realtà, le entità hanno caratteristiche

diverse, contengono una quantità diversa di dati, ecc. Di conseguenza è necessario adottare una vista multi-dimensionale.

Per permettere una visualizzazione multi-dimensionale dei dati, sono state sviluppate tecniche, conosciute anche come schemi e sono:

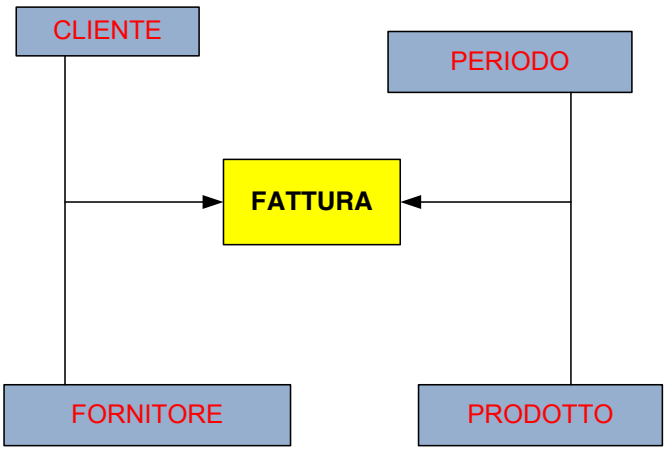
- *Star schema (schema a stella)*
- *Snowflake schema (schema a fiocco di neve)*
- *Mixed schema (schema misto)*

8.1 Star schema

Un modello che utilizza la tecnica a stella riflette il modo in cui un utente vede i dati.

Ad es. i dati contenuti in una fattura attraverso le dimensioni cliente, prodotto, fornitore, periodo.

Figura 6 - Esempio di Star Schema



La **tabella dei fatti** (fattura) contiene solo attributi che misurano il business oltre agli identificativi (le chiavi esterne).

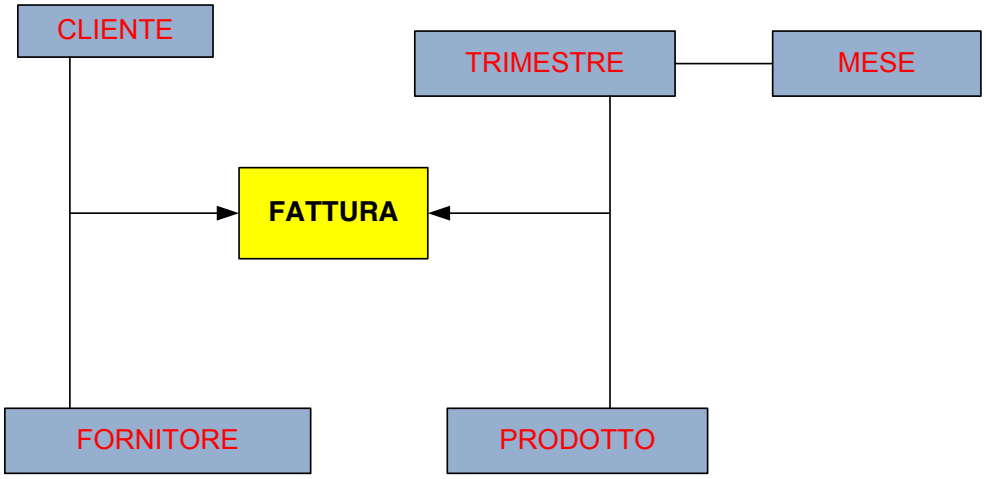
Le **tabelle dimensioni** (cliente, prodotto, fornitore, periodo) contengono attributi che descrivono la dimensione, oltre agli identificativi (chiavi) che indicizzano e organizzano i dati della tabella dei fatti.

8.2 Snowflake schema

È un'estensione del modello a stella nel quale una o più punte della stella si estendono. Tutte le tabelle dimensione sono normalizzate.

Ad es. la tabella della dimensione *periodo* viene normalizzata in *trimestre* e in *mese*.

Figura 7 - Esempio di Snowflake schema



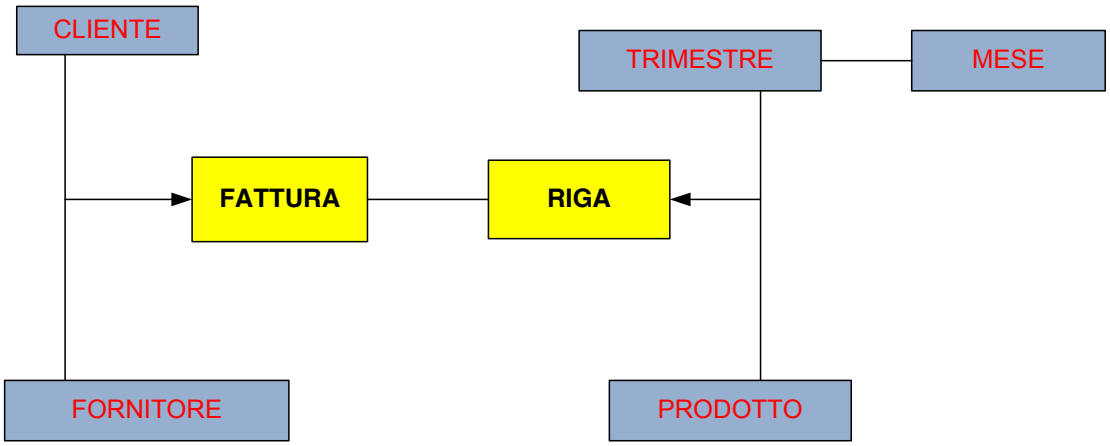
I vantaggi rispetto alla tecnica a stella sono un miglioramento delle prestazioni delle interrogazioni attraverso una migliore occupazione di spazio dovuta all'eliminazione dei dati ridondanti e l'utilizzo di piccole tabelle normalizzate invece di grosse tabelle non normalizzate.

Gli svantaggi sono dovuti al fatto che presenta una struttura più complessa, vi è una creazione di un maggior numero di tabelle e quindi maggiore difficoltà nella decisione di quale tabella utilizzare in una query.

8.3 Mixed schema

In alcune situazioni vi potrebbero essere delle tabelle dimensione con differenze sostanziali nel numero di attributi e nel volume. In questo caso non si può utilizzare il modello a stella o quello a fiocco di neve per tutta la struttura: bisogna quindi ricorrere ad una combinazione dei due, detta modello misto.

Figura 8 - Esempio di Mixed schema



La decisione su quale modello di dati utilizzare dipende dalle caratteristiche dei dati e dai requisiti richiesti dall'organizzazione che utilizzerà il Data Warehouse.

9. Gestione dei dati e ottimizzazione delle prestazioni

Un Data Warehouse contiene una quantità rilevante di dati. Il suo successo dipende quindi anche dall'efficienza con cui essi vengono gestiti. Vi sono tecniche di ottimizzazione delle prestazioni che vengono utilizzate per permettere al Data Warehouse di soddisfare le esigenze degli utenti. Queste tecniche contribuiscono a ridurre lo spazio su disco necessario a contenere i dati e a migliorare i tempi di risposta.

Si definisce *granularità* il livello di dettaglio presente in una unità di dati. Tale livello di dettaglio corrisponde al livello al quale i dati vengono registrati nel Data Warehouse. La granularità ha impatti sul volume dei dati mantenuti e sul tipo di query che potrebbe essere realizzata. Esistono tre livelli di granularità:

- **Alto** = i dati sono aggregati e sommarizzati, perciò conservati in un formato meno dettagliato. Riduce lo spazio necessario e migliora il tempo di risposta, ma impone limiti ai tipi di query in quanto i dati sono meno dettagliati.
- **Basso** = i dati sono conservati in maniera molto dettagliata.
- **Doppio** = certi dati sono conservati in maniera dettagliata e altri vengono sommarizzati e aggregati.

Un Data Warehouse può essere ottimizzato con determinate tecniche:

- **Partizionamento** = consiste nel dividere una unità di dati in unità più piccole secondo criteri definiti a priori. La ripartizione dei dati in gruppi più piccoli facilita inoltre la gestione e l'accesso.
- **Sommarizzazione** = crea un livello alto di granularità: riduce il livello di dettaglio e quindi assiste gli utenti nel processo decisionale soprattutto per quanto riguarda le previsioni. Non esistono regole fisse per la sommarizzazione, dipendono dai bisogni specifici della analisi richieste. I vantaggi sono in termini di necessità di minor spazio su disco, tempi di risposta più rapidi e costi di gestione inferiori. L'unico svantaggio è la perdita di dettaglio dei dati.

Capitolo 3

Progettazione e Gestione di un Data Warehouse

1. Identificazione delle esigenze

Quando si costruisce un Data Warehouse è necessario identificare e soddisfare i bisogni degli utenti finali, dei progettisti e degli sviluppatori. Progettisti e sviluppatori saranno in grado di costruire un Data Warehouse sulla base dei requisiti espressi dagli utenti finali.

Un utente finale utilizza il Data Warehouse come supporto al processo decisionale. Le esigenze possono essere classificate in 3 categorie:

- **Interrogazione (Query)** = le esigenze di un utente finale in termini di interrogazioni possono essere determinate attraverso la raccolta di esempi di possibili queries dai vari settori dell'azienda.
- **Analisi** = tutti i tipi di analisi che un utente finale potrebbe essere portato a fare devono essere identificati.
- **Reportistica** = le esigenze in termini di reportistica possono essere determinate attraverso la raccolta di esempi di tutti i reports prodotti dai vari settori dell'azienda.

2. Il compito del progettista

È responsabile della definizione topologica del Data Warehouse in considerazione delle esigenze espresse dagli utenti. Sulla base di tali esigenze il progettista decide sull'ampiezza del Data Warehouse, la quantità di dati che devono essere conservati, sia storici che attuali, e la tecnologia che deve essere utilizzata, sia hardware che software.

3. Il compito dello sviluppatore

È responsabile della conversione del modello concettuale (logico) dei dati in un modello logico (fisico). Deve inoltre stabilire:

- Dove definire fisicamente le basi dati e le tabelle
- Le applicazioni che, all'interno di ciascun componente, eseguono i processi necessari all'espletamento delle richieste degli utenti
- Il linguaggio di programmazione più appropriato per le applicazioni
- I protocolli di comunicazione
- Un metodo e degli strumenti efficienti di accesso ai dati
- I metodi di ottimizzazione e sicurezza dei dati

4. La progettazione di un Data Warehouse

Progettare un Data Warehouse significa selezionare le fonti dei dati di cui un'azienda ha bisogno e definire le strutture dei dati che costituiranno il suo Data Warehouse.

Dopo aver identificato le esigenze, il progettista deve scegliere le fonti da cui estrarre i dati. Uno dei compiti più importanti è assicurarsi che le fonti contengano dati accurati, completi, aggiornati.

Le fonti dati di un Data Warehouse sono classificabili in 3 categorie:

- **Interni** = i dati interni sono i dati generati all'interno dell'azienda e registrati sui sistemi transazionali utilizzati dall'azienda per la propria gestione.
- **Archiviati** = i dati archiviati sono dati non utilizzati al momento da parte dell'azienda, e che sono stati spostati su supporti magnetici remoti. Questi dati costituiscono una base storica molto importante su cui appoggiare le analisi di tendenza nel corso del supporto alle decisioni.
- **Esterni** = vengono generati al di fuori dell'azienda e sono dati di tipo commerciale: forniscono informazioni di tipo concorrenziale e di tendenza del mercato.

Dopo aver scelto le fonti di dati per il Data Warehouse i progettatori definiscono le strutture dei dati. Definiscono quindi le entità che vengono mappate sulle tabelle, e rappresentano le relazioni esistenti tra le entità utilizzando la modellazione a stella (STAR), a fiocco di neve (SNOWFLAKE) o miste (MIXED).

I progettatori convertono le definizioni logiche delle strutture in definizioni fisiche attraverso la creazione di basi di dati. Durante questo processo le entità vengono mappate su tabelle, e vengono altresì definiti gli attibuti e i campi chiave di queste ultime. Vengono anche definiti degli attributi supplementari che indicano la temporalità (per esempio, data e ora in cui il record è stato scritto)
ed altri che conterranno dati aggregati o sommarizzati. Questo processo di conversione viene talvolta definito come la "mappatura dal modello concettuale (logico) al modello logico (fisico)".

Le strategie di gestione dei dati sono:

- **de normalizzazione** = la tecnica di tenere i dati su un minor numero di tabelle di grandezza superiore
- **indicizzazione** = la classificazione dei dati in modo da fornire un rapido accesso
- **partizionamento** = frazionamento di una singola unità di dati in un numero di unità più piccole
- **aggregazione** = la sommarizzazione dei dati.

Il percorso di implementazione di un Data Warehouse comprende le seguenti tappe:

- creare le interfacce tra il Data Warehouse e le fonti di dati = permettono al Data Warehouse di essere popolato con i dati delle fonti.
- raffinare i dati attraverso l'utilizzo di strumenti di pulitura e di trasformazione = comprende la pulizia e la trasformazione dei dati.
    - *Pulizia* = viene svolta attraverso programmi di utilità che possono essere acquistati sul mercato o sviluppati internamente.
    - *Trasformazione* = aggiunta di un elemento "tempo" ai dati; aggregazione dei dati.
- caricare i dati sul Data Warehouse = popolamento del Data Warehouse.
- mettere il Data Warehouse a disposizione degli utenti = significa completare l'installazione e il collaudo. Viene consegnato agli utenti a sezioni e non in blocco unico. Questo approccio è più sicuro in quanto si possono verificare i feedback di una sezione e sulla base di questi caricare poi le successive sezioni.

Dopo che il Data Warehouse è stato messo a disposizione degli utenti è possibile che sorga il bisogno di apportarvi delle modifiche. Di tanto in tanto con frequenza prestabilita i dati in esso devono essere aggiornati (refreshed).

Fase cruciale è poi anche quella della manutenzione, dove bisogna gestire le dimensioni del Data Warehouse, assicurare la sicurezza dei dati in esso

contenuti e monitorare le sue prestazioni (la prestazione è misurata attraverso i tempi di risposta alle interrogazioni).

L'accesso ai dati infatti per sua natura comporta dei rischi. Un accesso totale ed indiscriminato a tutti i dati aziendali non è concepibile proprio a causa dei rischi ad esso connessi. Si introduce quindi un concetto di dare agli utenti dei *diritti di accesso*.

Capitolo 4

Implementazione di un DW in azienda

1. Data Warehouse aziendale e Data Mart

Di centrale importanza per un'azienda è l'implementazione di un Data Warehouse o di DM. Questi ultimi sono sostanzialmente una sorta di banca dati specializzata che è destinata a sostenere analisi di business e contenere i dati da una o diverse fonti.

La differenza tra un DM e un Data Warehouse è il loro campo di applicazione. Un Data Mart cerca solo di soddisfare le esigenze di una parte della società, come il marketing o dipartimento delle finanze. Può essere pensato come un Warehouse di piccole dimensioni.

Un Data Warehouse cerca di servire l'intera azienda. Quindi viene definito *aziendale* quando risponde alle esigenze commerciali dell'intera azienda, e cioè contiene dati aziendali che vengono utilizzati da tutti i settori all'interno dell'azienda per prendere delle decisioni sul futuro della stessa.

La creazione di **molteplici Data Mart settoriali (approccio buttom up)** in alternativa ad un unico Data Warehouse aziendale comporta la frammentazione nella visione d'insieme dei dati di un'azienda. Si andranno a costruire Data Mart per ogni settore di utenti all'interno dell'azienda, e dalla loro combinazione si perviene al Data Warehouse. Risulterà di conseguenza difficile ricomporre a posteriori queste *viste* per ottenere la visione globale dell'andamento dell'azienda. Per rendere l'idea l'autore William Inmon disse:

" *Si possono catturare tutti i pesciolini del mare, comunque non si avrebbe una balena*"

Quello che dice in sostanza è che i vari Data Mart avranno dei divari, mai offriranno la vista del livello aziendale a livello globale. La scelta di un Data Mart in alternativa del Data Warehouse potrebbe essere dettata dall'esigenza di implementare un sistema di supporto delle decisioni in un breve arco di tempo e con risorse ridotte.

Un **Data Warehouse centralizzato (approccio top down)** approccio che spinge a costruire il Data Warehouse in primis considerando le esigenze di tutta l'azienda nella sua progettazione. Viene utilizzato per conservare i dati di un organizzazione in un unico contenitore in quanto la maggioranza dei processi decisionali viene svolta in un unico luogo. È ovviamente un

compito molto complesso bisogna considerare e studiare come l'intera organizzazione funziona. Con il deposito una volta costruito, è possibile creare data mart che non sarebbero altro che l'estratto di una porzione di *magazzino dati* per soddisfare le esigenze di settori specifici, quali la finanza o marketing.

È consigliato quando:

- i dati sono integrati all'interno dell'organizzazione e la sede centrale richiede una visione globale di essi
- una singola base dati soddisfa le esigenze di memorizzazione dei dati di tutta l'azienda.
- I dati non sono facilmente accessibili se vengono tenuti nelle varie sedi.

Vi è poi un'alternativa chiamata **Client/Server**. Vari computers su cui sono conservati gli spezzoni del Data Warehouse vengono considerati o come clients o come servers.

Clients sono i computers che richiedono un servizio da un altro computer. Servers sono i computers che forniscono un servizio ad un altro computer. Client e server possono talvolta risiedere nello stesso computer.

Vi sono due alternative possibili:

a due livelli (Client/Server puro) = esistono due livelli di piattaforma tecnica. I servizi client vengono tenuti su una delle piattaforme, i servizi server su un'altra. Sia il posto di lavoro che il mainframe possono essere entrambi utilizzati con la funzione server. Gli strumenti di accesso ai dati sono tenuti sul client.

a tre livelli (Mainframe/server/posto di lavoro) = tre livelli di piattaforma tecnica. Il Mainframe viene utilizzato per l'estrazione e la trasformazione dei dati provenienti dalle fonti di dati. Il server contiene i dati del Data Warehouse e il software relativo ai componenti dello stesso. Il posto di lavoro contiene le applicazioni per gli utenti (front-end) ed i dati eventualmente estratti dal Data Warehouse (DM).

2. Registrazione dei dati

Un Data Warehouse richiede diverse unità, logiche e fisiche, su cui tenere e con cui gestire i dati. A causa dell'alto volume di dati che fanno parte di un Data Warehouse, la gestione deve essere efficace ed efficiente riguardo i tempi e i costi. La scelta tecnologica deve orientarsi verso l'hardware e i software che permettono l'utilizzo di una varietà di mezzi. Un solo tipo di unità disponibile (es. Direct Access Storage Device – DASD) non è sufficiente per un Data Warehouse maturo.

Le principali unità da tenere in considerazione sono:

- *Memoria centrale* = costo elevato, accesso estremamente rapido
- *Memoria estesa* = costo relativamente elevato, accesso estremamente rapido
- *DASD* = costo abbordabile, accesso molto rapido
- *Altri supporti magnetici* = costo basso, accesso abbastanza lento.

3. Implementazione del Data Warehouse in azienda

Il Data Warehouse è un sistema informativo dove i dati sono organizzati e strutturati per un facile accesso da parte dell'utente e per fornire supporto ai processi decisionali. Sono abilitati i seguenti sistemi:

- **DSS (Decisional Support System)** = utilizzato per risolvere problemi specifici
- **EIS (Executive/Enterprise Information System)** = consente una continua circolazione dei dati non dipendente da problemi specifici

Nelle banche e in generale nelle istituzioni finanziarie gli ambiti di utilizzo sono molteplici, poiché tutte le aree gestionali di tali organizzazioni sono caratterizzate da volumi considerevoli di dati su cui devono essere prese decisioni strategiche. Poiché il Data Warehouse può avere un valore strategico, all'interno di tali tipi di organizzazioni è fondamentale per il management definire una strategia. La strategia per il Data Warehouse è essenzialmente un percorso evolutivo che porta l'azienda da applicazioni Data Warehouse *non "mission-critical"* verso una situazione in cui il Data

Warehouse è una componente fondamentale del sistema informativo aziendale.

La strategia di data warehousing di un'azienda può essere classificata in base a due dimensioni fondamentali:

1. **utilizzo del DATA WAREHOUSE esistente**: livello di maturità degli utenti e delle funzioni di supporto del Data Warehouse nell'utilizzo dell'esistente;
2. **utilizzo del DATA WAREHOUSE in prospettiva**: di utilizzo del Data Warehouse come piattaforma di decision support.

Le aziende attraversano dunque quattro fasi nella storia dell'utilizzo del Data Warehouse:

- la prima fase, chiamata **supporto** (basso utilizzo del Data Warehouse esistente, basso utilizzo prospettico del Data Warehouse), è la fase in cui si trovano le aziende che hanno fallito uno o più progetti di Data Warehousing e non pensano di ampliarne l'utilizzo prospettico. In questa fase si possono trovare anche aziende che non hanno un Data Warehouse e non pensano di realizzarlo;
- la seconda fase, chiamata **opportunità** (basso utilizzo del Data Warehouse esistente, alto utilizzo prospettico del Data Warehouse), è la fase in cui si trovano le aziende che, pur avendo fallito uno o più progetti di warehousing o avendo semplicemente esplorato la tematica senza approfondirla, puntano a sviluppare le attività di decision support tramite il Data Warehouse.
- la terza fase (alto utilizzo del Data Warehouse esistente, alto utilizzo prospettico del Data Warehouse), è quella fase in cui il Data Warehouse diviene **strategico** per i processi decisionali aziendali. In questa fase si trovano tutte quelle aziende che hanno intrapreso con successo un progetto di warehousing e che ne stanno sfruttando a pieno le potenzialità;
- la quarta fase, chiamata **factory** (alto utilizzo del Data Warehouse esistente, basso utilizzo prospettico del Data Warehouse) è la fase in cui si trovano le aziende in cui il Data Warehouse è maturo, la metodologia di implementazione consolidata e le aree decisionali

critiche sono presidiate. In questa fase l'imperativo principale è l'efficienza e il risparmio di costi derivanti dal Data Warehouse e dal suo utilizzo. Un processo di sclerotizzazione nell'uso del Data Warehouse può in alcuni casi far tornare l'azienda alla prima fase.

3. Come si possono superare i limiti del sistema basato su database relazionali?

Da quello che si è riscontrato, con il preesistente sistema basato su database relazionali non è possibile effettuare delle analisi più complesse, altrimenti s'incapperebbe in ritardi temporali nell'ordine di ore. Inoltre non è possibile effettuare analisi di dati nelle loro varie sfaccettature, nel senso che, quando si cercano di intrecciare più variabili tra di loro il sistema va in crisi.

Questi limiti sono superati grazie ad uno strumento (DW) che permette di effettuare un'aggregazione di dati a monte, per cui risulterà più efficiente effettuare analisi più complesse, sia in termini di fattibilità che in termini di tempi di risposta. Le analisi sui dati che si possono eseguire con un DW sono chiamate OLAP. Mentre con un sistema di query reporting l'utilizzatore può chiedere, ad esempio, qual è il prodotto più venduto in un particolare mese dell'anno, in una particolare area geografica, con un'analisi di tipo OLAP si può chiedere al sistema di effettuare un report di tutti i prodotti più venduti, in tutti i mesi e in tutte le aree geografiche. Permette di effettuare interrogazioni contemporaneamente su più dimensioni. Per questo viene visto come un ipercubo, dovuto al fatto che vi

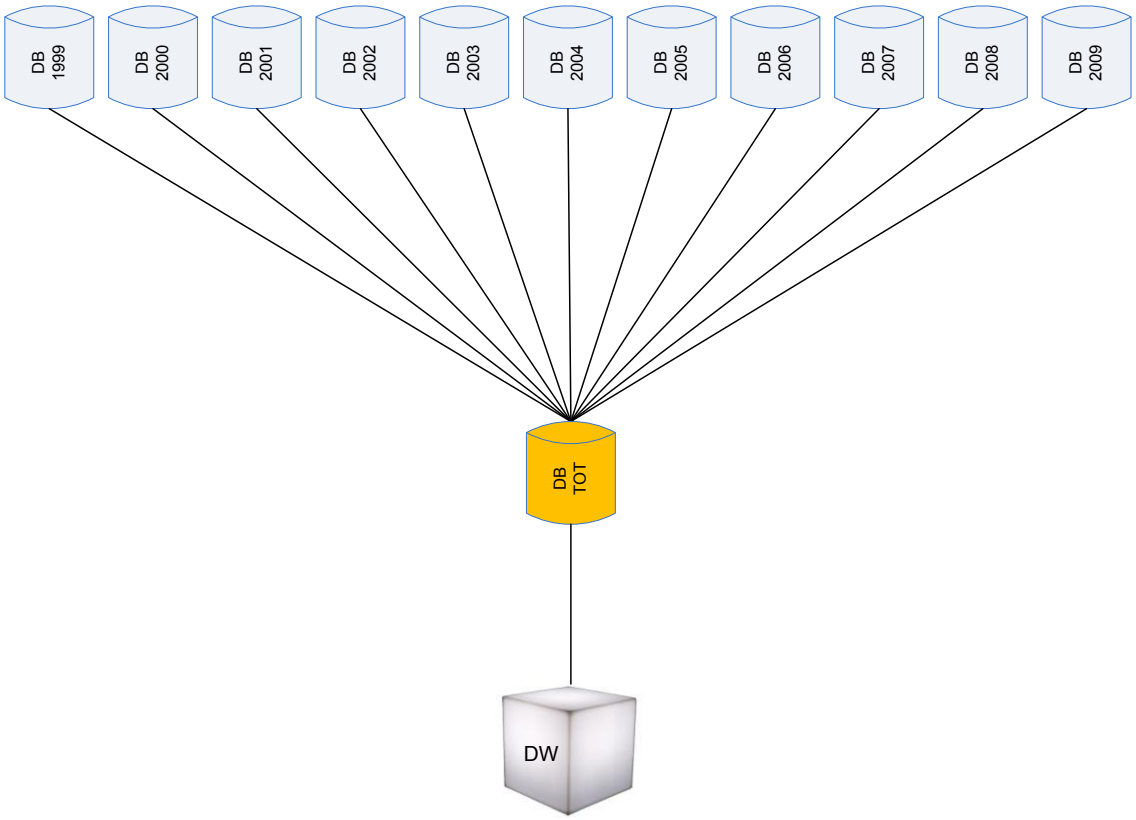
è un intreccio di più dimensioni come quella geografica, temporale, ecc. Inoltre, a seconda di chi è il decision maker, si interrogherà il cubo in un ottica differente. Ad es. il manager regionale, responsabile delle vendite in una particolare regione, sarà interessato solo alla vendita dei prodotti in un determinato mercato; mentre il manager finanziario esaminerà la vendita dei prodotti in tutti i mercati relativamente al periodo corrente e quello precedente; il manager di prodotto esamina la vendita di un prodotto in tutti i periodi e in tutti i mercati; infine il manager strategico si concentrerà su una categoria di prodotti, un'area regionale e un orizzonte temporale medio. Un sistema basato su Data Warehouse permette, quindi, analisi multi reporting, analisi più complesse e più utili a fini commerciali.

4. Progettazione DW per essere popolato dai dati presenti nei database relazionali

Dopo aver constatato i limiti di un sistema basato su database relazionali ed aver chiarito come un DW li affronta con successo, passiamo alla sua progettazione sulla base delle tabelle presenti nel sistema preesistente. Innanzitutto dobbiamo considerare il fatto che il sistema di database relazionali presentava per ogni singolo anno un database diverso. Nel DW dovranno confluire tutti i dati in un'unica struttura. Per far ciò due sono le strade percorribili:

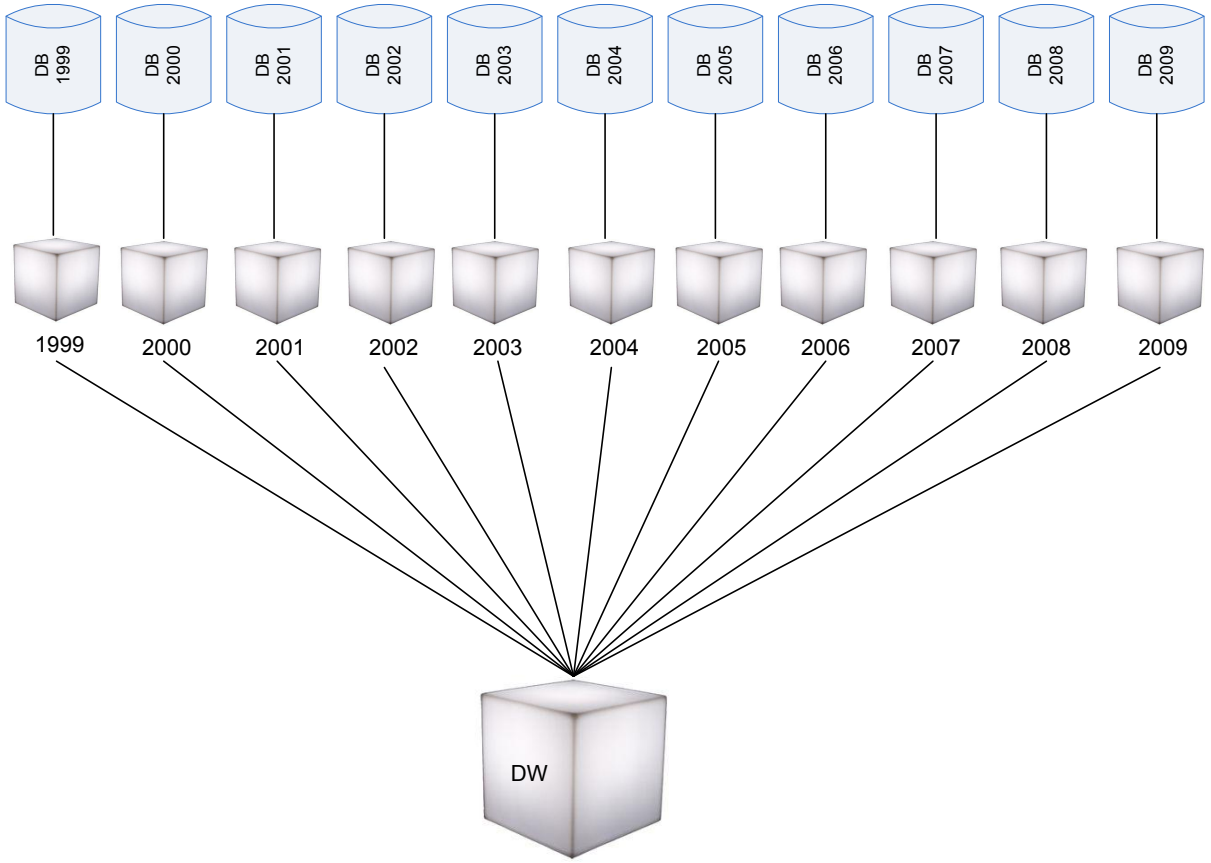
1. Far confluire tutti i dati relativi ai vari database dei singoli anni in un unico database che li contenga tutti (*fig. 18*).

Figura 18 - Popolamento DW caso 1



2. Creare un cubo dati che risulterà come la somma di tante partizioni di cubo, consistenti ciascuna di queste, ad un anno di gestione della cooperativa (*fig.19*).

Figura 19- Popolamento DW caso 2



La strada che ho preferito percorrere è la prima, ossia far confluire tutti i dati dei database relativi ai vari anni di gestione in un unico database per poi elaborarlo come un cubo dati.

Prima però di procedere con la migrazione dei dati nel database totale, dobbiamo operare ad una pulizia dei dati, ossia l'eliminazione dei campi ritenuti irrilevanti e l'aggiunta di altri indispensabili per l'elaborazione del nostro cubo. In primis è avvenuta la creazione dei campi "anno" e "mese" che saranno calcolate come la parte della data relativa all'anno e al mese, il che ci servirà per la dimensione temporale. Successivamente sono state create delle colonne calcolate per eseguire:

- il totale delle vendite ($Qtò\ evasa \cdot Prezzo\ Vendita$)
- il totale costo standard ($Qtò\ evasa \cdot Costo\ standard$)
- il totale costo ultimo ($Qtò\ evasa \cdot Costo\ ultimo$)

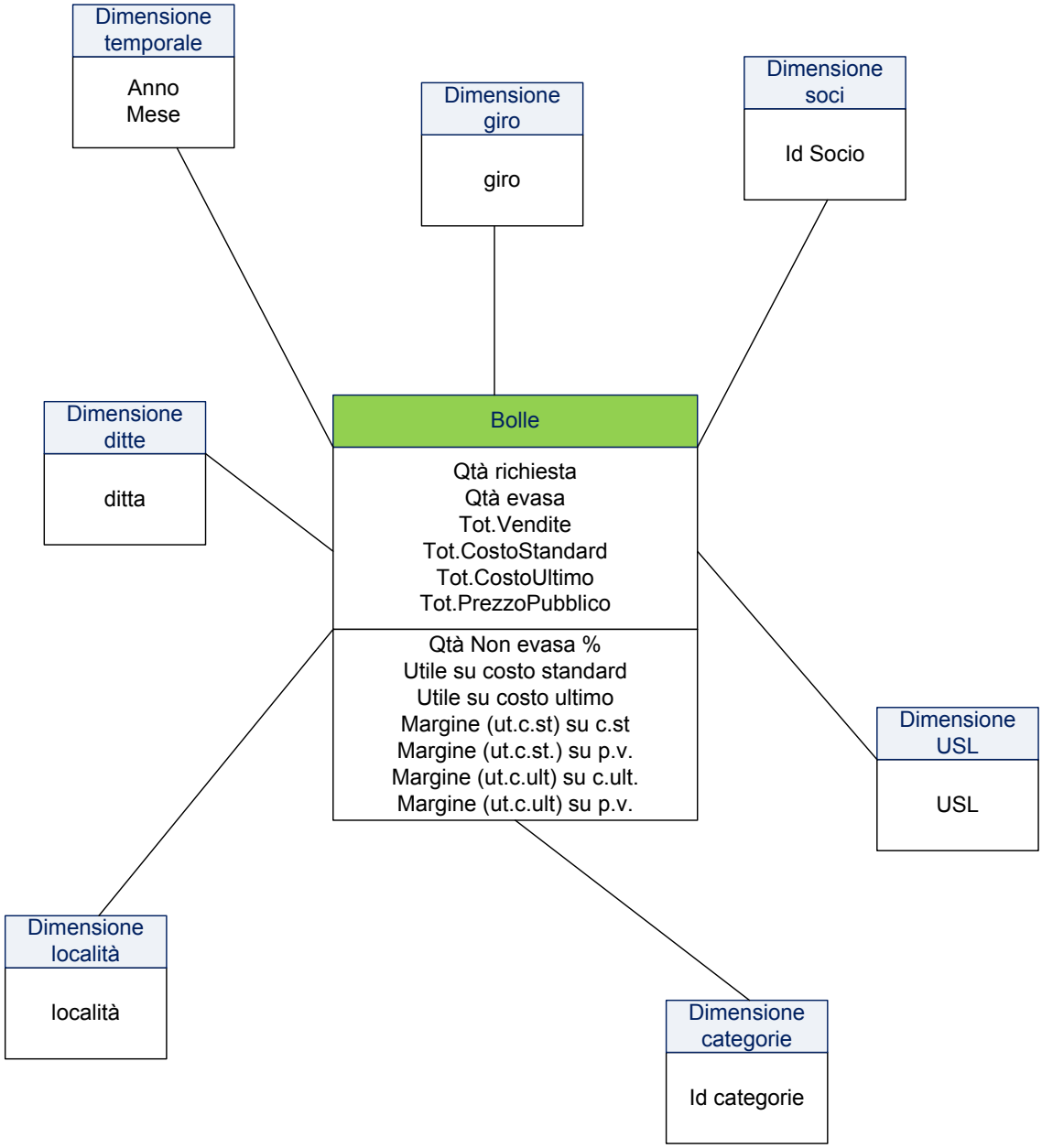
- il totale prezzo pubblico ($Qt\grave{o}\ evasa \cdot Prezzo\ Pubblico$)

Fatto questo si è potuto popolare il database totale con tutti i dati presenti nei database "annuali". Dopodiché possiamo passare alla progettazione vera e propria del cubo dati. Viene stabilita come **tabella dei fatti,** la tabella "bolle". Si selezionano come misure del nostro cubo *"Qtà richiesta", "Qtà evasa", "Totale vendite", Totale costo standard", "Totale costo ultimo", "Totale prezzo pubblico"*, quindi, come si può facilmente intuire, tutti campi misurabili. Per quanto riguarda, invece, le **tabelle di dimensione**, sono state così disposte:

- dimensione temporale suddivisa per anno e per mese
- dimensione soci
- dimensione località
- dimensione categorie
- dimensione giro
- dimensione ditte
- dimensione USL

Tutte le dimensioni sono state create con modello ***star schema***.

Figura 20 - Progettazione cubo dati



Infine una volta creata la struttura del nostro cubo sono stati generati dei membri calcolati, come la quantità non evasa percentuale, l'utile su costo standard e su costo ultimo e i vari margini.

Lo strumento che ho utilizzato per l'implementazione di questo cubo dati così progettato è Microsoft SQL Server 2000 – Analysis Services.

5. Creazione del cubo dati in SQL Server – Analysis Services

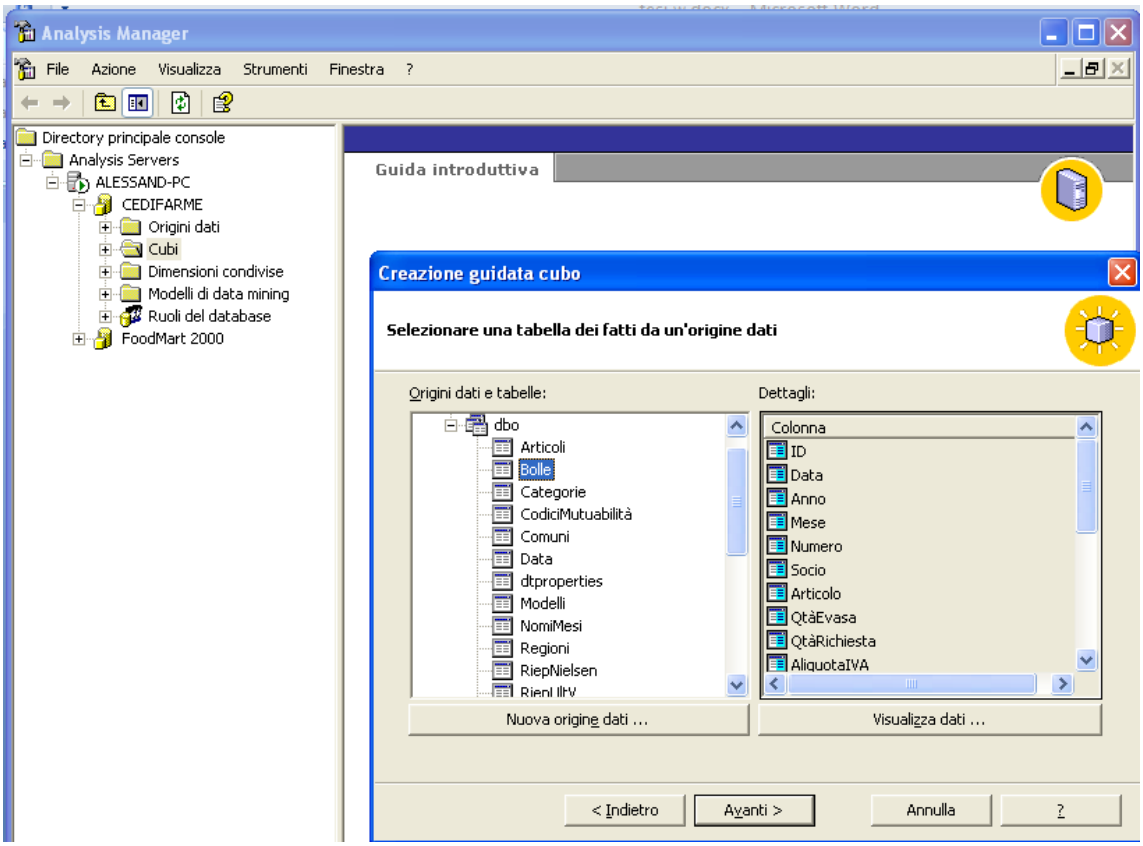
Finita la progettazione del DW si procede con la sua creazione in SQL server 2000 Analysis services. Analizzerò molto sinteticamente questa fase perché una fase meramente tecnica e che può cambiare in base allo strumento che si utilizza per creare il DW. Di vitale importanza è la progettazione del DW, perché se fatta bene potrà essere implementata con qualsiasi strumento e senza problemi.

Il primo step prevede la realizzazione dell'origine di dati, ossia far attingere i dati ad SQL Server da quelli presenti nei DB operazionali. Per far ciò, si apre lo strumento Analysis Manager incluso nel pacchetto installativo di SQL server 2000 e si fa click tasto desto del mouse sul nome del server sul quale stiamo lavorando e in seguito "Nuovo database". Appena inserito il nome abbiamo creato il Database su cui lavorare. Successivamente clicchiamo nuovamente tasto destro del mouse su origine dati e poi su "Nuova origine dati". In questo modo possiamo selezionare l'origine dati da cui il nostro DW andrà a popolarsi una volta elaborato.

Step successivo è quello di creare il nostro cubo dati. Con la stessa procedura esposta per l'origine dati, cliccare su "Cubi" e poi su "Nuovo cubo". Fatto questo si attiverà una procedura guidata di creazione cubo, dove sarà possibile impostare le misure e le dimensioni dello stesso. Noi lo faremo così come è stato impostato in fase di progettazione.

Si sceglie come tabella dei fatti la tabella bolle e si prosegue con la procedura guidata.

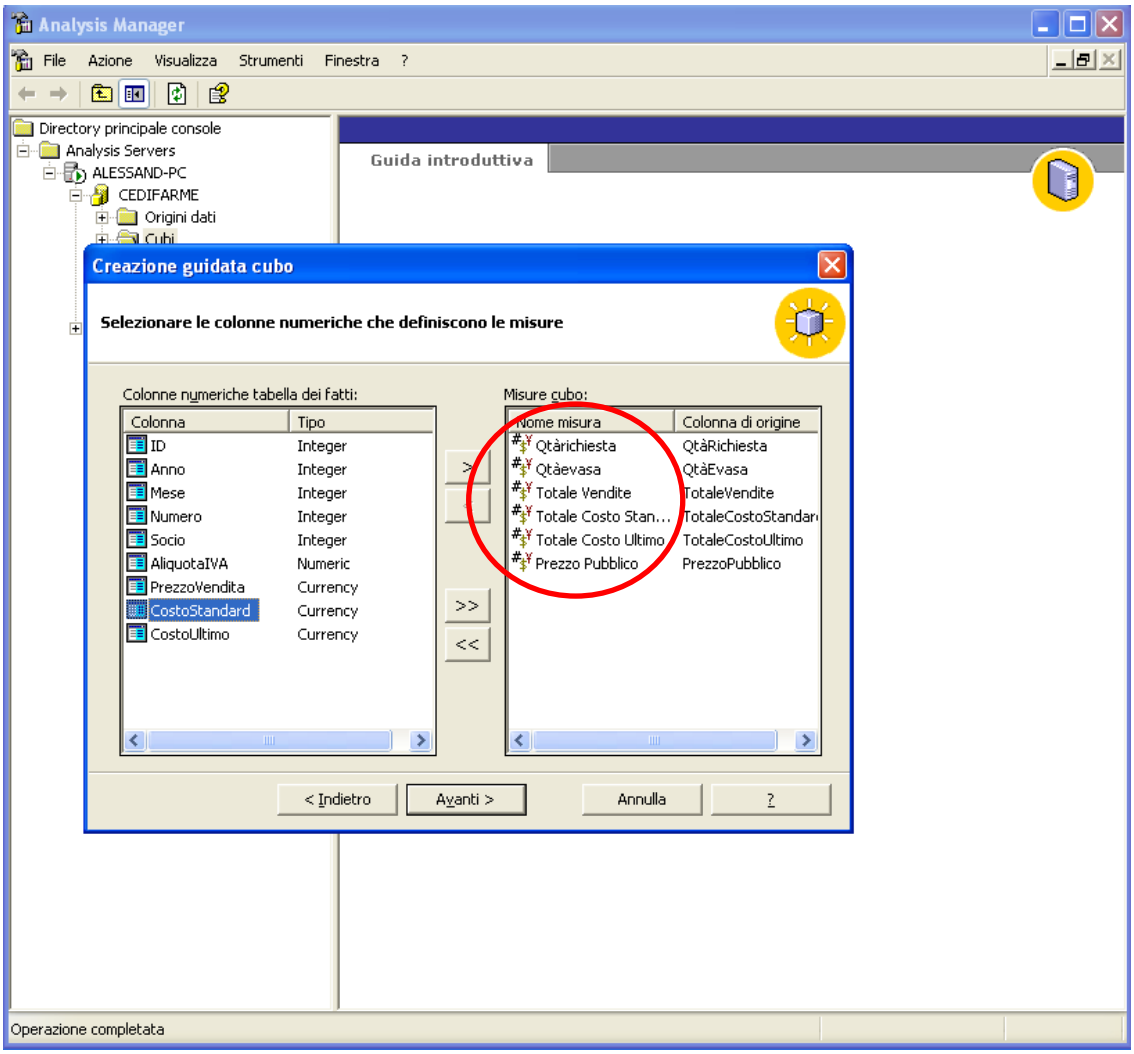
Figura 21 - creazione guidata cubo: selezionare tabella dei fatti



Dopodiché verranno selezionate le misure del cubo che nel nostro caso saranno:

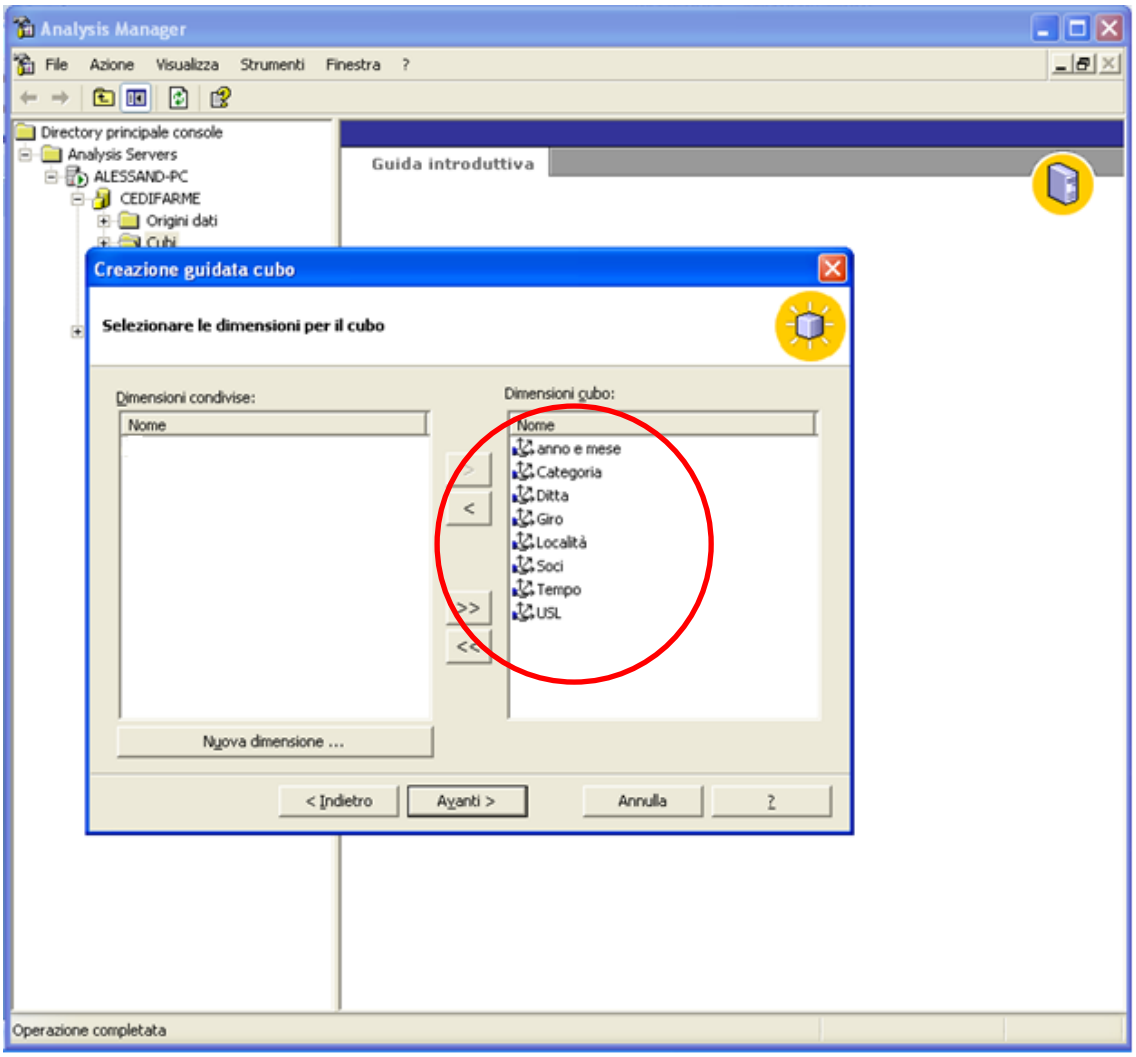
- quantità richiesta
- quantità evasa
- tot. Vendite
- tot. Costo standard
- tot. Costo ultimo
- tot. Prezzo pubblico

Figura 22 - creazione guidata cubo: selezionare le misure



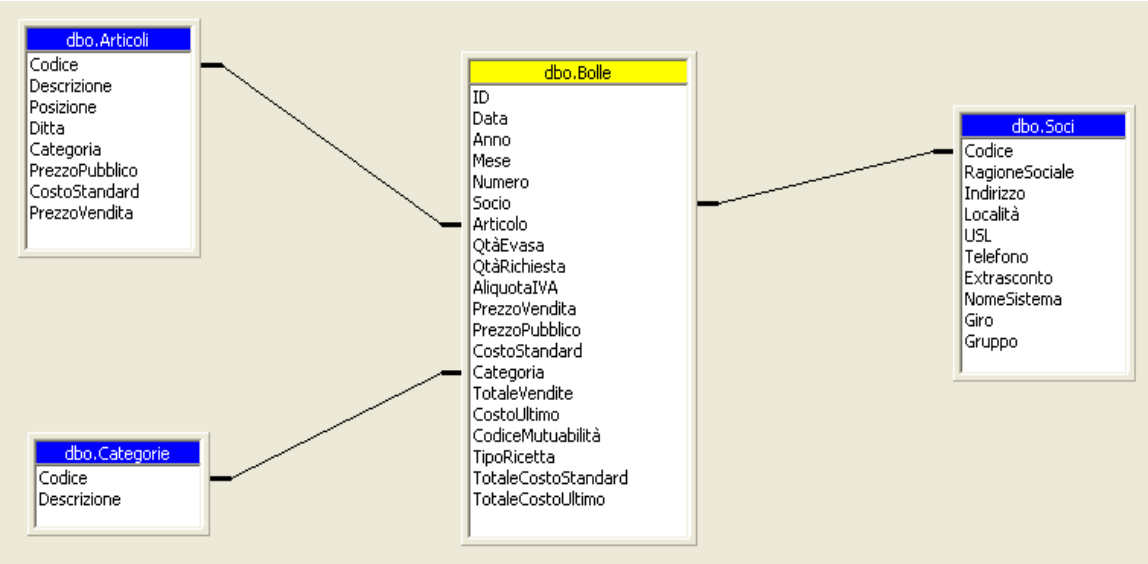
Fatto questo si passa alla creazione delle dimensioni, che sono quelle decise in fase di progettazione.

Figura 23 - creazione guidata cubo: selezionare le dimensioni



Concluse queste operazioni si è creato il nostro cubo il quale può essere visualizzato nella sua struttura tramite l'editor cubi.

Figura 24 - Editor cubi



Nell'editor cubi è possibile modificare o aggiungere dimensioni e misure del cubo che abbiamo creato.

6. Visualizzazione dei dati presenti nel cubo

Il primo strumento importantissimo per l'esplorazione dei dati presenti nel DW è il visualizzatore cubi già presente nel pacchetto analysis services di SQL server 2000. Da qui è possibile effettuare drill up (quando si vuole considerare un livello di aggregazione più alto dei dati) e drill down (per esplodere i dati in livelli di dettaglio più ampi).

Supponiamo di voler esaminare il totale delle vendite effettuate nel 2009 per ogni categoria di prodotto. Inseriamo nelle *facce* del cubo la dimensione temporale e quella delle categorie.

Figura 25 - visualizzatore cubi

	+ Anno	
Categoria	- Totale Tempo	+ 2009
- Totale Categoria	€ 53.869.731,66	€ 53.869.731,66
2	€ 1.451.885,63	€ 1.451.885,63
3	€ 297.137,19	€ 297.137,19
4	€ 3.967.407,21	€ 3.967.407,21
5	€ 2.440.044,06	€ 2.440.044,06
6	€ 2.582.937,26	€ 2.582.937,26
7	€ 400.856,92	€ 400.856,92
A	€ 41.693.515,95	€ 41.693.515,95
B	€ 1.035.947,44	€ 1.035.947,44

Il totale tempo presenta gli stessi risultati del totale del 2009, perché, come accennato in precedenza, per semplicità di analisi abbiamo inserito nel DW solo i dati inerenti al 2009. Da questa analisi si visualizza il totale delle vendite nel 2009 per ogni categoria di prodotto.

Se volessimo esplodere i nostri dati a livello mensile basterà effettuare un semplice drill down sulla dimensione temporale.

Figura 26 - visualizzatore cubi: drill down categoria e tempo

- Anno		- 2009												
Mese	Totale 2009	gennaio	febbraio	marzo	aprile	maggio	giugno	luglio	agosto	settembre	ottobre	novembre	dicembre
Categoria														
- Totale Categoria	€ 53.869.731,66	€ 4.687.545,69	€ 4.232.798,53	€ 4.690.391,60	€ 4.521.538,78	€ 4.475.487,20	€ 4.530.115,12	€ 4.555.636,54	€ 3.904.662,72	€ 4.595.672,23	€ 4.752.949,25	€ 4.460.539,04	€ 4.462.394,96
2	€ 1.451.885,63	€ 147.311,57	€ 121.912,26	€ 126.564,29	€ 120.415,50	€ 117.764,12	€ 111.270,70	€ 115.944,53	€ 117.538,22	€ 117.795,88	€ 123.063,47	€ 112.990,73	€ 119.534,36
3	€ 297.137,19	€ 19.746,89	€ 18.846,32	€ 20.091,49	€ 27.509,00	€ 33.621,93	€ 30.993,09	€ 31.270,87	€ 25.721,92	€ 25.504,23	€ 22.165,02	€ 20.602,91	€ 21.063,52
4	€ 3.967.407,21	€ 330.783,67	€ 297.031,82	€ 335.609,17	€ 326.308,11	€ 321.661,72	€ 330.983,52	€ 348.573,43	€ 317.803,55	€ 350.688,42	€ 347.984,63	€ 338.878,50	€ 321.100,67
5	€ 2.440.044,06	€ 214.938,95	€ 190.576,00	€ 202.190,77	€ 202.434,66	€ 184.855,62	€ 210.727,93	€ 213.205,07	€ 193.484,48	€ 215.818,77	€ 216.427,75	€ 197.971,18	€ 197.412,88
6	€ 2.582.937,26	€ 214.984,87	€ 204.644,00	€ 222.438,99	€ 214.082,69	€ 224.820,05	€ 233.101,28	€ 246.260,58	€ 210.516,46	€ 219.295,64	€ 212.267,27	€ 184.460,39	€ 196.065,04
7	€ 400.856,92	€ 34.213,92	€ 29.959,41	€ 34.934,45	€ 31.541,08	€ 34.110,52	€ 31.595,68	€ 32.930,82	€ 28.888,93	€ 33.414,56	€ 36.145,70	€ 39.559,64	€ 33.562,21
A	€ 41.693.515,95	€ 3.619.468,98	€ 3.279.918,38	€ 3.657.429,17	€ 3.514.251,09	€ 3.477.670,78	€ 3.504.762,69	€ 3.492.571,79	€ 2.932.099,85	€ 3.549.012,00	€ 3.698.670,65	€ 3.473.494,06	€ 3.494.166,51
B	€ 1.035.947,44	€ 106.096,84	€ 89.910,34	€ 91.343,27	€ 84.996,65	€ 80.992,46	€ 76.680,23	€ 74.879,45	€ 78.609,31	€ 84.142,73	€ 96.224,76	€ 92.581,63	€ 79.489,77

Già possiamo riscontrare la semplicità e la rapidità rispetto ad un sistema basato su DB relazionale. I tempi di risposta e le complessità di interrogazioni sono di gran lunga più efficienti ed efficaci.

Nel precedente esempio è stata utilizzata come misura il totale delle vendite. Ovviamente possono essere impostate altre dimensioni che abbiamo inserito nel cubo dati, ed è anche possibile impostarne più di una, il che molto utile per eseguire dei confronti. Anche le dimensioni possono variare tra quelle che abbiamo impostato in sede di creazione del cubo, e possono essere utilizzate più di una nella visualizzazione che stiamo facendo.

7. Utilizzo di Microsoft Excell come strumento di reporting

Implementato il cubo dati in Microsoft SQL Server non ci resta che effettuare le possibili analisi dei dati presenti in esso. Lo strumento che ho utilizzato per far ciò è Microsoft Excel (versione 2007) il quale permette di realizzare un *collegamento dati* utile per esplorare e studiare i dati presenti nel DW.

Una volta eseguito il collegamento con il DW si potranno scegliere le dimensioni che si intende visualizzare nella tabella pivot, le misure e i filtri. I tempi di risposta sono piuttosto insignificanti in confronto alle query eseguite sul sistema a DB relazionale. Infatti si aggirano intorno alla decina di secondi per effettuare anche esplorazioni molto complesse, come ad esempio incrociando più dimensioni con più misure e considerando l'intero ammontare dei dati senza utilizzare alcun filtro. In sostanza qualunque sia l'interrogazione che vogliamo esigere dal sistema non ci porterà via più di dieci secondi prima di rispondere.

Inoltre, una caratteristica importante di Microsoft Excel è quella di poter eseguire le più disparate operazioni sui dati ottenuti dal DW, applicando operatori e funzioni già presenti nell'applicazione. È anche possibile sviluppare una macro quando si vuole, ad esempio, una certa ricorrenza nell'esecuzione di determinate procedure.

Detto questo passerò ad illustrare una panoramica di interrogazioni fatte sul DW che riprendono in maniera del tutto similare quelle proposte sul sistema relazionale utilizzato dal manager delle vendite di CEDIFARME soc. coop.

7.1 Tabulato vendite per socio a quantità

Come si può facilmente intuire questo report illustra le quantità vendute nel tempo (espresso in anni e mesi) ripartite tra i vari soci e classificate per le diverse categorie di prodotti venduti. Per un motivo di privacy ho inserito l'ID del socio anziché il proprio nominativo.

Per eseguire questa analisi è bastato semplicemente selezionare le dimensioni temporale, del socio e delle categorie, utilizzando come misura la quantità venduta. Il sistema ha avuto tempi di risposta, come già accennato, di qualche secondo.

In questo caso ho selezionato tutti i soci, tutte le categorie e tutti i mesi dell'anno 2009, ma ovviamente ai fini di un'analisi più mirata, può essere selezionato solo uno o un gruppo di soci e/o categorie. In automatico il sistema mi fornisce anche i totali e, inoltre, può anche restituire valori minimi, massimi, medi, oppure si potrebbe ordinare i dati (in ordine crescente o decrescente) con lo scopo, ad esempio, di mostrare una top list di chi ha venduto di più in un periodo di tempo determinato.

Tabella 1 - Tabulato vendite per socio a quantità

tempo-soci/categorie	2	3	4	5	6	7	A	B	Totale complessivo
2009	*801.555*	*29.492*	*882.448*	*1.224.144*	*449.449*	*146.281*	*9.458.627*	*333.658*	*13.325.654*
gennaio	84.999	2.033	77.522	110.130	38.097	14.354	874.530	40.940	*1.242.605*
100	420	3	271	443	61	3	3.641	187	*5.029*
1002	177	7	56	303	50	29	2.221	33	*2.876*
1004	331	5	185	260	90	26	2.241	84	*3.222*
1006	128	1	240	699	81	30	3.708	112	*4.999*
1008	114	9	496	67	134	45	5.871	237	*6.973*
1009	149	31	237	162	69	44	3.375	78	*4.145*
1010	230	9	107	116	64	25	2.012	77	*2.640*
				[...]					
dicembre	56.913	2.001	70.862	84.023	31.888	10.126	774.690	23.698	1.054.201
100	403	2	316	447	61	7	4.570	182	5.988
1002	102	7	61	228	56	14	1.921	31	2.420
1004	258	2	210	275	62	35	2.154	66	3.062
1006	93	4	260	217	49	9	4.100	73	4.805
1007	12						12		24
1008	133	3	384	94	82	19	4.439	60	5.214
1009	49	16	175	49	60	33	2.419	40	2.841
1010	164	1	39	80	27	91	1.821	38	2.261
				[...]					
Totale complessivo	***801.555***	***29.492***	***882.448***	***1.224.144***	***449.449***	***146.281***	***9.458.627***	***333.658***	***13.325.654***

7.2 Tabulato vendite per socio a quantità e valore

La seguente tabella, mostra i risultati in termini non più di sole quantità vendute, ma ci dà indicatori di valore. Quest'ultimo è inteso come totale delle vendite e dei costi (costo standard e costo ultimo).

Oltre a questo è presentata, in termini percentuali, la cosiddetta "rottura di magazzino", ossia gli ordini che non sono stati evasi nonostante fossero stati richiesti. I dati sono raggruppati per anno, socio e categoria. Questa tabella risulterebbe molto utile, ad esempio per conoscere il valore creato da ogni singolo socio e quale categoria emerge come "di punta". Anche in questo caso, come il precedente e i successivi, è possibile restringere il

campo a determinati soci, categorie o un determinato periodo temporale di riferimento.

Tabella 2 - Tabulato vendite per socio a quantità e valore

tempo-categorie-soci/qtà-valore	Qtàrichiesta	Qtàevasa	non evasa percent.	Totale Vendite	Totale Costo Standard	Totale Costo Ultimo
2009	**13.713.058**	**13.325.654**	**2,83%**	**€ 94.650.037,22**	**€ 87.426.493,80**	**€ 86.470.865,06**
100	*61.260*	*59.808*	*2,37%*	*€ 385.195,41*	*€ 364.523,06*	*€ 360.143,83*
2	3.636	3.633	0,08%	€ 14.268,94	€ 12.476,98	€ 12.581,56
3	9	9	0,00%	€ 89,54	€ 80,62	€ 81,86
4	4.216	4.128	2,09%	€ 15.474,74	€ 13.739,97	€ 13.852,85
5	5.108	5.139	-0,61%	€ 24.503,08	€ 21.751,41	€ 21.806,58
6	969	969	0,00%	€ 5.252,87	€ 4.497,60	€ 4.557,04
7	144	144	0,00%	€ 481,45	€ 289,27	€ 290,41
A	44.533	43.555	2,20%	€ 316.533,66	€ 303.430,77	€ 298.579,48
B	2.645	2.231	15,65%	€ 8.591,13	€ 8.256,44	€ 8.394,05
1002	*32.452*	*32.317*	*0,42%*	*€ 221.651,13*	*€ 199.282,16*	*€ 195.650,70*
2	1.231	1.231	0,00%	€ 5.407,11	€ 4.493,57	€ 4.640,95
3	93	92	1,08%	€ 1.019,00	€ 850,49	€ 882,08
4	1.003	1.000	0,30%	€ 5.365,88	€ 4.520,79	€ 4.561,47
5	4.433	4.437	-0,09%	€ 21.197,51	€ 18.140,09	€ 18.122,92
6	617	616	0,16%	€ 3.907,09	€ 3.192,80	€ 3.237,07
7	417	409	1,92%	€ 973,24	€ 695,56	€ 740,14
A	24.276	24.150	0,52%	€ 181.837,83	€ 165.631,15	€ 161.666,33
B	382	382	0,00%	€ 1.943,47	€ 1.757,71	€ 1.799,74
1004	*38.525*	*37.875*	*1,69%*	*€ 239.206,07*	*€ 222.719,99*	*€ 220.053,56*
2	3.038	3.040	-0,07%	€ 13.134,63	€ 11.700,36	€ 11.945,73
3	107	107	0,00%	€ 1.275,29	€ 1.141,57	€ 1.162,24

[...]

7.3 Tabulato vendite per località

Quest'ulteriore report ci mostra i risultati in termini di quantità evasa ripartiti per anni e per località. Quindi si considera anche l'aspetto geografico e si suddividono i dati per categorie. Un'interrogazione molto utile per conoscere dove si effettuano più vendite di prodotti e quale categoria si vende maggiormente in un determinato luogo geografico.

Tabella 3 - Tabulato vendite per località

tempo - località/categorie	2	3	4	5	6	7	A	B	Totale complessivo
2009	**801.555**	**29.492**	**882.448**	**1.224.144**	**449.449**	**146.281**	**9.458.627**	**333.658**	**13.325.654**
ADELFIA	401	8	234	952	274	26	4.203	159	6.257
AIELLO DEL SABATO	988	55	951	1.458	643	122	10.199	342	14.758
ALTAMURA	2.327	27	2.091	5.136	1.397	349	18.972	556	30.855
ALTAVILLA IRPINA	2.593	210	1.925	2.699	990	256	23.406	1.105	33.184
ANDRETTA	42			107	6	2	84	9	250
ANDRIA	58.763	1.245	62.087	94.837	34.336	10.487	787.965	26.881	1.076.601
APICE	2.151	125	1.075	2.534	968	182	14.179	555	21.769
APRICENA	8.904	300	13.291	15.560	5.267	1.725	134.061	3.312	182.420
AQUILONIA	98		66	186	62	13	1.140	51	1.616
ARIANO IRPINO	3.988	207	6.448	4.776	2.045	844	42.356	1.430	62.094
ATELLA	1.986	188	1.283	2.698	684	136	37.373	1.334	45.682
ATRIPALDA				40		2			42
AVELLINO	4.555	178	3.341	8.674	2.549	555	62.871	2.168	84.891
BAGNOLI IRPINO	647	47	432	1.525	293	152	15.632	199	18.927
BARI	66.731	2.032	56.162	74.758	34.496	10.225	628.961	25.088	898.453
				[...]					
ZAPPONETA	2.876	97	4.880	7.217	2.127	1.127	51.733	1.474	71.531
ZUNGOLI	1.719	98	847	2.302	762	225	22.588	732	29.273
Totale complessivo	*801.555*	*29.492*	*882.448*	*1.224.144*	*449.449*	*146.281*	*9.458.627*	*333.658*	*13.325.654*

7.4 Tabulato venduto annuale per categoria

Nella tabella seguente è illustrata un'analisi relativa a statistiche (quantità evasa, totale vendite, totale costo standard, utile su costo standard, margini su tale utile, totale costo ultimo, utile su costo ultimo e margini su tale utile) relative alle categorie di prodotti venduti divise per tempo espresso in anni. È possibile comunque effettuare il drill-down sulla dimensione tempo per visualizzare questi dati anche ad un livello temporale che si estende alla mensilità.

Tabella 4 - Tabulato venduto annuale per categoria

tempo-categorie/misure	Qtàevasa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2009	**13.325.655**	**€ 94.650.175,22**	**€ 87.426.618,80**	**€ 7.223.556,42**	**8,26%**	**7,63%**
2	801.555	€ 3.388.016,17	€ 3.089.325,47	€ 298.690,70	9,67%	8,82%
3	29.492	€ 325.843,82	€ 285.024,92	€ 40.818,90	14,32%	12,53%
4	882.448	€ 5.309.189,41	€ 4.626.472,06	€ 682.717,35	14,76%	12,86%
5	1.224.144	€ 6.029.737,18	€ 5.250.257,22	€ 779.479,96	14,85%	12,93%
6	449.449	€ 3.049.227,79	€ 2.571.832,38	€ 477.395,41	18,56%	15,66%
7	146.282	€ 504.338,15	€ 326.703,40	€ 177.634,75	54,37%	35,22%
A	9.458.627	€ 74.459.672,84	€ 69.797.413,82	€ 4.662.259,02	6,68%	6,26%
B	333.658	€ 1.584.149,86	€ 1.479.589,53	€ 104.560,33	7,07%	6,60%
Totale complessivo	***13.325.655***	***€ 94.650.175,22***	***€ 87.426.618,80***	***€ 7.223.556,42***	***8,26%***	***7,63%***

tempo-categorie/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.
2009	**€ 86.470.990,06**	**€ 8.179.185,16**	**9,46%**	**8,64%**
2	€ 3.170.835,99	€ 217.180,18	6,85%	6,41%
3	€ 291.066,33	€ 34.777,49	11,95%	10,67%
4	€ 4.664.205,39	€ 644.984,02	13,83%	12,15%
5	€ 5.270.372,77	€ 759.364,41	14,41%	12,59%
6	€ 2.613.007,63	€ 436.220,16	16,69%	14,31%
7	€ 337.945,78	€ 166.392,37	49,24%	32,99%
A	€ 68.609.110,71	€ 5.850.562,13	8,53%	7,86%
B	€ 1.514.445,46	€ 69.704,40	4,60%	4,40%
Totale complessivo	**€ 86.470.990,06**	**€ 8.179.185,16**	**9,46%**	**8,64%**

7.5 Statistica utile cooperativa per socio

Qui la statistica (stesse misure che sono state utilizzate per l'interrogazione precedente) non è presentata per sole categorie negli anni, ma viene aggiunta un'ulteriore informazione (dimensione) relativa ai soci. Come già accennato precedentemente è possibile limitare l'interrogazione a determinati soci o categorie.

Tabella 5 - Statistica utile cooperativa per socio

tempo - categoria - soci/misure	Qtàevasa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2009	**13.325.655**	**€94650175,22**	**€87426618,8**	**€7223556,42**	**8,26%**	**7,63%**
2	*801.555*	*€3388016,17*	*€3089325,47*	*€298690,7*	*9,67%*	*8,82%*
100	3.633	€14268,94	€12476,98	€1791,96	14,36%	12,56%
1002	1.231	€5407,11	€4493,57	€913,54	20,33%	16,90%
1004	3.040	€13134,63	€11700,36	€1434,27	12,26%	10,92%
1006	1.502	€6797,43	€6098,61	€698,82	11,46%	10,28%
1007	12	€24	€29,76	€-5,76	-19,35%	-24,00%
1008	1.167	€5322,86	€4504,02	€818,84	18,18%	15,38%
1009	1.137	€5329,99	€4749,91	€580,08	12,21%	10,88%
1010	2.285	€9858,06	€8300	€1558,06	18,77%	15,80%
			[...]			
B	*333.658*	*€1584149,86*	*€1479589,53*	*€104560,33*	*7,07%*	*6,60%*
100	2.231	€8591,13	€8256,44	€334,69	4,05%	3,90%
1002	382	€1943,47	€1757,71	€185,76	10,57%	9,56%
1004	771	€3606	€3441,23	€164,77	4,79%	4,57%
1006	971	€4664,24	€4199,14	€465,1	11,08%	9,97%
1008	1.273	€6621,96	€6241,14	€380,82	6,10%	5,75%
1009	1.009	€5081,79	€4863,11	€218,68	4,50%	4,30%
1010	579	€2784,23	€2475,33	€308,9	12,48%	11,09%

tempo - categoria - soci/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.
2009	**€86470990,06**	**€8179185,16**	**9,46%**	**8,64%**
2	*€3170835,99*	*€217180,18*	*6,85%*	*6,41%*
100	€12581,56	€1687,38	13,41%	11,83%
1002	€4640,95	€766,16	16,51%	14,17%
1004	€11945,73	€1188,9	9,95%	9,05%
1006	€6209,06	€588,37	9,48%	8,66%
1007	€29,76	€-5,76	-19,35%	-24,00%
1008	€4714,95	€607,91	12,89%	11,42%
1009	€4782,54	€547,45	11,45%	10,27%
1010	€8430,07	€1427,99	16,94%	14,49%
		[...]		
B	*€1514445,46*	*€69704,4*	*4,60%*	*4,40%*
100	€8394,05	€197,08	2,35%	2,29%
1002	€1799,74	€143,73	7,99%	7,40%
1004	€3516,47	€89,53	2,55%	2,48%
1006	€4350,94	€313,3	7,20%	6,72%
1008	€6385,37	€236,59	3,71%	3,57%
1009	€4946,09	€135,7	2,74%	2,67%
1010	€2568,74	€215,49	8,39%	7,74%

7.6 Tabulato venduto per categoria e data

In questo caso il riepilogo non è fatto innanzitutto per la dimensione temporale, bensì per categoria. Si può così studiare ad esempio il trend delle misure oggetto di analisi, nel tempo. In questo caso abbiamo solamente l'anno 2009 per semplicità di studio, ma in realtà il DW presenta i dati relativi agli ultimi 11 anni e quindi questo tipo di osservazione sarebbe più significativa.

Tabella 6 - Tabulato venduto per categoria e data

categoria - data/misure	Qtàevasa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2	**801.555**	**€3388016,17**	**€3089325,47**	**€298690,7**	**9,67%**	**8,82%**
2009	801.555	€3388016,17	€3089325,47	€298690,7	9,67%	8,82%
3	**29.492**	**€325843,82**	**€285024,92**	**€40818,9**	**14,32%**	**12,53%**
2009	29.492	€325843,82	€285024,92	€40818,9	14,32%	12,53%
4	**882.448**	**€5309189,41**	**€4626472,06**	**€682717,35**	**14,76%**	**12,86%**
2009	882.448	€5309189,41	€4626472,06	€682717,35	14,76%	12,86%
5	**1.224.144**	**€6029737,18**	**€5250257,22**	**€779479,96**	**14,85%**	**12,93%**
2009	1.224.144	€6029737,18	€5250257,22	€779479,96	14,85%	12,93%
6	**449.449**	**€3049227,79**	**€2571832,38**	**€477395,41**	**18,56%**	**15,66%**
2009	449.449	€3049227,79	€2571832,38	€477395,41	18,56%	15,66%
7	**146.281**	**€504200,15**	**€326578,4**	**€177621,75**	**54,39%**	**35,23%**
2009	146.281	€504200,15	€326578,4	€177621,75	54,39%	35,23%
A	**9.458.627**	**€74459672,84**	**€69797413,82**	**€4662259,02**	**6,68%**	**6,26%**
2009	9.458.627	€74459672,84	€69797413,82	€4662259,02	6,68%	6,26%
B	**333.658**	**€1584149,86**	**€1479589,53**	**€104560,33**	**7,07%**	**6,60%**
2009	333.658	€1584149,86	€1479589,53	€104560,33	7,07%	6,60%
Totale complessivo	**13.325.654**	**€94650037,22**	**€87426493,8**	**€7223543,42**	**8,26%**	**7,63%**

categoria - data/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.
2	**€3170835,99**	**€217180,18**	**6,85%**	**6,41%**
2009	€3170835,99	€217180,18	6,85%	6,41%
3	**€291066,33**	**€34777,49**	**11,95%**	**10,67%**
2009	€291066,33	€34777,49	11,95%	10,67%
4	**€4664205,39**	**€644984,02**	**13,83%**	**12,15%**
2009	€4664205,39	€644984,02	13,83%	12,15%
5	**€5270372,77**	**€759364,41**	**14,41%**	**12,59%**
2009	€5270372,77	€759364,41	14,41%	12,59%
6	**€2613007,63**	**€436220,16**	**16,69%**	**14,31%**
2009	€2613007,63	€436220,16	16,69%	14,31%
7	**€337820,78**	**€166379,37**	**49,25%**	**33,00%**
2009	€337820,78	€166379,37	49,25%	33,00%
A	**€68609110,7**	**€5850562,13**	**8,53%**	**7,86%**
2009	€68609110,7	€5850562,13	8,53%	7,86%
B	**€1514445,46**	**€69704,4**	**4,60%**	**4,40%**
2009	€1514445,46	€69704,4	4,60%	4,40%
Totale complessivo	**€86470865,1**	**€8179172,16**	**9,46%**	**8,64%**

7.7 Statistica utile cooperativa per categoria e ditta

Questa query è simile alle precedenti con la sola variante che, mentre prima era selezionata come dimensione del cubo dati quella dei soci, qui è stata impostata quella delle ditte. Quindi si possono analizzare le statistiche di cui sopra, per ogni anno (o determinati periodi temporali immessi dall'utente) per ogni categoria (o determinate categorie, anche una sola) ed infine per le ditte.

Tabella 7 - Statistica utile cooperativa per categoria e ditta

tempo-categoria-ditta/misure	Qtàevasa	Totale Vendite	Totale Costo Standard	Utile su costo standard	Margine (utile su c.st.) su c.st.	Margine (utile su c.st.) su p.v
2009	**13.325.654**	**€ 94.650.037**	**€ 87.426.494**	**€ 7.223.543**	**8,26%**	**7,63%**
2	*801.555*	*€ 3.388.016*	*€ 3.089.325*	*€ 298.691*	*9,67%*	*8,82%*
1002	1	€ 10	€ 7	€ 3	36,34%	26,65%
1070	184	€ 604	€ 357	€ 247	69,03%	40,84%
1071	612	€ 2.598	€ 2.106	€ 491	23,32%	18,91%
1211	2.865	€ 16.658	€ 14.315	€ 2.343	16,37%	14,07%
1269	318	€ 1.258	€ 1.049	€ 209	19,95%	16,63%
1598	73	€ 308	€ 175	€ 133	75,77%	43,11%
162	742	€ 5.161	€ 3.841	€ 1.320	34,37%	25,58%
1646	3.904	€ 22.536	€ 20.266	€ 2.270	11,20%	10,07%
				[...]		
B	*333.658*	*€ 1.584.150*	*€ 1.479.590*	*€ 104.560*	*7,07%*	*6,60%*
1070	23	€ 70	€ 31	€ 39	126,05%	55,76%
1071	2.963	€ 19.616	€ 18.661	€ 956	5,12%	4,87%
1116	1.091	€ 6.537	€ 6.336	€ 201	3,17%	3,07%
1269	666	€ 3.902	€ 3.747	€ 155	4,14%	3,98%
1286	176	€ 1.201	€ 757	€ 444	58,69%	36,98%
1314	18.357	€ 139.398	€ 124.789	€ 14.609	11,71%	10,48%
1598	4.710	€ 28.811	€ 27.267	€ 1.544	5,66%	5,36%
1660	132	€ 1.192	€ 1.126	€ 66	5,84%	5,52%
1686	8.494	€ 78.288	€ 74.476	€ 3.812	5,12%	4,87%
				[...]		

tempo-categoria-ditta/misure	Totale Costo Ultimo	Utile su costo ultimo	Margine (utile su c.ult.) su c.ult.	Margine (utile su c.ult.) su p.v.
2009	**€ 86.470.865**	**€ 8.179.172**	**9,46%**	**8,64%**
2	*€ 3.170.836*	*€ 217.180*	*6,85%*	*6,41%*
1002	€ 7	€ 3	36,34%	26,65%
1070	€ 357	€ 247	69,03%	40,84%
1071	€ 2.158	€ 440	20,39%	16,94%
1211	€ 14.315	€ 2.343	16,37%	14,07%
1269	€ 1.113	€ 145	13,00%	11,50%
		[...]		
B	*€ 1.514.445*	*€ 69.704*	*4,60%*	*4,40%*
1070	€ 31	€ 39	126,05%	55,76%
1071	€ 18.661	€ 956	5,12%	4,87%
1116	€ 6.334	€ 202	3,19%	3,09%
1269	€ 3.841	€ 62	1,61%	1,59%
1286	€ 1.125	€ 76	6,78%	6,35%
1314	€ 140.202	-€ 803	-0,57%	-0,58%
		[...]		

Come si è potuto constatare le analisi fatte sui dati presenti nel DW sono del tutto similari a quelle fatte tramite interrogazioni sui Database relazionali. Gli svantaggi che quest'ultimo sistema presenta sono, come più volte esposto, il tempo di risposta molto lungo e la quasi impossibilità di eseguire delle query più complesse, altrimenti si rischia di mandarlo in timeout. Mentre abbiamo notato come le analisi effettuate sul DW sono state tutte rivolte all'intera popolazione dei dati presenti in esso, nel senso che sono stati selezionati tutti i "membri" delle dimensioni impostate come "facce" del cubo. Le problematiche di un sistema basato su database relazionali sono state risolte senza effettuare un investimento significativo in elaboratori che riescono ad eseguire le operazioni più velocemente, i quali rischierebbero di non risolverle affatto.