Un Quadro Metodologico per la Costruzione e l'Uso di un Data Warehouse*

Luca Cabibbo Riccardo Torlone

Dipartimento di Informatica e Automazione, Università di Roma Tre
Via della Vasca Navale, 79 — I-00146 Roma, Italy
E-mail: {cabibbo,torlone}@dia.uniroma3.it

Sommario In tutte le applicazioni delle basi di dati in ambiti commerciali e di servizio sta emergendo sempre di più la necessità dello sviluppo e della gestione di *data warehouse*, magazzini storico-temporali di dati aziendali dedicati ad analisi orientate al supporto alle decisioni strategiche da intraprendere. Esistono oggi numerosi strumenti che realizzano molte delle attività connesse alla costruzione e alla gestione di questi magazzini; tuttavia, la tecnologia esistente non è sostenuta da opportune metodologie (se non per attività specifiche) in grado di guidare il progettista nell'intero ciclo di vita di un data warehouse.
In questo lavoro viene proposto un quadro metodologico di carattere generale, che suggerisce e coordina tutte le attività necessarie alla costruzione e l'uso di un data warehouse, a partire dalla selezione delle sorgenti informative, fino all'organizzazione di collezioni di dati orientate all'analisi multidimensionale. Alcune fasi di questo quadro corrispondono a metodologie consolidate, per le quali esistono strumenti efficienti dedicati alla loro realizzazione; per altre fasi vengono proposte metodologie nuove, basate su modelli di dati originali.

1 Introduzione

È ormai ben noto che l'analisi storico-temporale dei dati operazionali offre alle imprese grandi opportunità per ottenere vantaggi in termini di competitività e di razionalizzazione nell'uso delle risorse. Ad esempio, l'identificazione di andamenti inaspettati nelle vendite può suggerire ad un'azienda commerciale l'opportunità per nuovi affari, mentre l'analisi dei profili d'utenza può essere utile ad una azienda di servizi per definire tariffe e promozioni. Un *data warehouse* è una collezione integrata e persistente di dati aziendali, orientata al supporto alle decisioni, che viene costruita per favorire queste attività di analisi [7, 13, 14]. Il termine *data warehousing* indica invece il processo di estrazione dei dati dalle varie sorgenti informative aziendali (basi di dati, sistemi legacy, file di vario genere) e di integrazione in un singolo data warehouse.

Un data warehouse viene mantenuto fisicamente separato dalle sorgenti informative, per diversi motivi. Innanzitutto, i sistemi operazionali elaborano dati di

* Lavoro finanziato parzialmente dal *CNR* e dal *MURST*.

dettaglio, mentre in genere non hanno bisogno di mantenere dati storici; d'altra parte, i data warehouse devono gestire dati storici, tipicamente in forma aggregata. Inoltre, le varie sorgenti informative possono essere eterogenee, mentre un data warehouse deve offrire una visione integrata dell'intero patrimonio informativo aziendale. Infine, le applicazioni operazionali, generalmente interattive, non devono essere appesantite dalle attività di analisi. Peraltro, i tradizionali sistemi di gestione di basi di dati sono ottimizzati per le attività *transazionali* (On-Line Transaction Processing o OLTP), caratterizzate da molte transazioni concorrenti, ognuna delle quali coinvolge spesso pochi record, mentre un data warehouse deve favorire attività *analitiche* (On-Line Analytical Processing o OLAP), caratterizzate da poche interrogazioni, basate su aggregazioni, che coinvolgono un grande numero di record [8].

In effetti, l'analisi dei dati non avviene direttamente sul data warehouse, ma piuttosto su speciali magazzini di dati, derivati dal warehouse e chiamati *data mart* o basi di dati *multidimensionali*. Il termine "multidimensionale" origina dal fatto che l'efficacia dell'analisi è legata anche alla capacità di descrivere e manipolare i dati secondo diverse prospettive chiamate, appunto, "dimensioni." Ad esempio, in un'impresa commerciale, l'analisi orientata al supporto alle decisioni risulta più efficace se le singole vendite sono organizzate secondo dimensioni quali la tipologia di prodotto, il tempo, la località geografica.[1]

Negli ultimi anni si è assistito ad un rapidissimo sviluppo del data warehousing, sia da un punto di vista applicativo che da un punto di vista tecnologico. Si pensi che il volume di affari previsto per il 1998 si aggira intorno agli 8 miliardi di dollari. Esistono molti strumenti in commercio che favoriscono le singole attività e che vengono venduti come moduli di estensione dei sistemi commerciali già esistenti (Oracle, Informix, DB2, ecc.) oppure come pacchetti applicativi dedicati (ad esempio, Red Brick e Essbase). Questi strumenti realizzano, dal punto di vista della creazione, attività come il filtering dei dati, il loro caricamento nel data warehouse e l'aggiornamento incrementale. Dal punto di vista dell'uso del warehouse esistono invece i cosiddetti sistemi OLAP, che permettono la specifica di analisi complesse basate tipicamente su aggregazioni lungo le varie dimensioni e la visualizzazione grafica dei risultati dell'analisi. Questi ultimi sistemi si suddividono in sistemi relazionali (ROLAP), che memorizzano i dati in tabelle relazionali, e in sistemi multidimensionali (MOLAP), che rappresentano e memorizzano i dati nella forma di matrici multidimensionali.

Facendo uso di questi, ma anche di strumenti tradizionali, sono state sviluppate negli ultimi anni moltissime applicazioni reali. Tuttavia, come spesso avviene, a questo fortissimo sviluppo tecnologico e applicativo non è corrisposto lo sviluppo di metodologie opportune. Oggi la costruzione e la gestione dei data warehouse avviene sulla base di processi empirici, supportati magari da strumenti efficienti e metodi che favoriscono la soluzione di aspetti specifici, mentre mancano criteri e strategie di carattere generale ai quali fare riferimento. Come

[1] Esiste in effetti un'altra importante modalità di analisi dei dati fattuali che consiste nella ricerca di similarità in transazioni operazionali (il cosiddetto data mining), che comunque non verrà affrontata in questo lavoro.

conseguenza, spesso la qualità dei prodotti delle analisi è ben al di sotto delle attese, e i costi di creazione e manutenzione dei data warehouse risultano più elevati delle previsioni.

L'ambizione di questo lavoro è proprio quello di provare a definire un quadro metodologico di riferimento in grado di guidare il progettista nelle varie attività da compiere lungo *tutto* il ciclo di vita di un data warehouse: dalla selezione delle sorgenti informative, alla costruzione del warehouse, fino alla definizione delle basi di dati multidimensionali. Si cercherà di definire un quadro sufficientemente generale, sia rispetto ai prodotti che ai sistemi in gioco, in maniera che possa essere utilizzato indipendentemente sia dal problema allo studio che dagli strumenti a disposizione. Esso è organizzato in un insieme di fasi, per alcune delle quali esistono metodologie consolidate (ad esempio, quelle di reverse engineering e di integrazione di schemi) nonché strumenti di supporto; per altre fasi verranno invece proposti modelli e strategie originali.

Il quadro metodologico si basa su quattro macro-fasi principali: (1) selezione delle sorgenti informative, (2) integrazione, (3) progettazione del data warehouse e (4) progettazione delle basi di dati multidimensionali. Ognuna di queste fasi è suddivisa in alcuni sotto-passi da effettuare in cascata, che possono essere a loro volta organizzati in azioni (non necessariamente sequenziali) che si basano su criteri di carattere generale. Verranno usati, per la descrizione dei dati di ingresso e dei prodotti delle varie fasi, modelli di dati diversi. In particolare, nella prima fase si farà riferimento a modelli logici dei dati tradizionali, usati descrivere le sorgenti informative preesistenti. Per la seconda e la terza fase si farà invece riferimento al modello Entità Relazione (o semplicemente E-R) utilizzando la notazione del testo [3], nonché il modello relazionale. Infine, nell'ultima fase si farà riferimento a un modello logico originale per dati multidimensionali, indipendente dai criteri di rappresentazione usati dai sistemi correnti (tabelle relazionali o matrici dimensionali), ma faremo vedere anche possibili implementazioni. In questo modo l'approccio proposto risulta essere generico rispetto agli strumenti, e comunque realizzabile con i prodotti disponibili sul mercato.

Il risultato finale offre un inquadramento generale delle attività connesse alla costruzione all'uso di un data warehouse, che non va ovviamente interpretato come un processo stringente: nei casi pratici, alcune fasi possono essere soppresse perché non necessarie o non realizzabili in pratica, mentre altre possono richiedere attività specifiche ulteriori che non sono riconducibili a criteri di carattere generale. Crediamo comunque che il quadro risultante possa costituire un'utile guida pratica per i progettisti di applicazioni di data warehousing.

Il resto del lavoro è organizzato come segue. Nel Paragrafo 2 verrà presentato l'intero quadro metodologico e verranno brevemente descritti i modelli di dati usati nella metodologia. Nel Paragrafo 3 verranno descritte le prime fasi della metodologia generale, mentre il Paragrafo 4 è dedicato all'ultima fase che costituisce il contributo più originale della nostra ricerca.

2 Il Quadro Metodologico e i Modelli Usati

2.1 Presentazione del Quadro Metodologico

Il quadro metodologico di riferimento per la costruzione e l'uso di Data Warehouse è illustrato in Figura 1. Il quadro prevede l'esecuzione di 4 macro-fasi principali, ognuna delle quali è suddivisa in sotto-fasi, con i relativi dati di ingresso e prodotti. A loro volta, le varie fasi possono essere suddivise in passi elementari, che suggeriscono azioni da compiere all'interno della specifica attività. Per semplicità, la presentazione delle varie fasi e passi assume che questi siano eseguiti in sequenza; in realtà, come spesso avviene in contesti metodologici complessi, alcune attività possono essere omesse, altre svolte in parallelo, altre infine evidenziare che passi precedenti debbano essere completati o rieseguiti.

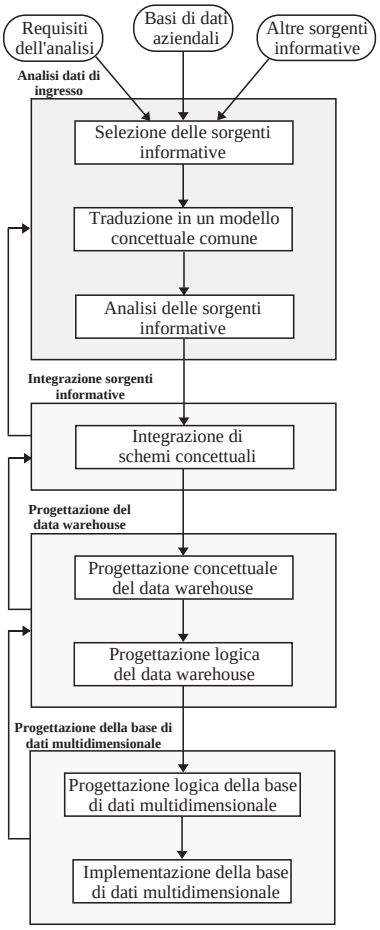

Figura1. Quadro metodologico di riferimento

Il quadro che si ottiene viene presentato sinteticamente nel seguito ed è preceduto dalla specifica dei dati di ingresso.

Dati di Ingresso. Lo svolgimento della metodologia richiede la specificazione di tre tipologie di informazioni di ingresso: (i) requisiti, (ii) schemi della sorgenti informative aziendali, (iii) schemi di altre sorgenti informativi disponibili. I requisiti sono una descrizione (solitamente in linguaggio naturale semistrutturato) delle esigenze aziendali di analisi. Gli schemi delle sorgenti informative aziendali descrivono formalmente la struttura delle basi di dati operative disponibili, nonché di altre eventuali sorgenti informative (ad esempio, di sistemi legacy), con la relativa documentazione di supporto (tra questa, il glossario aziendale dei termini). Inoltre, spesso accade che l'analisi dei dati aziendali richieda la correlazione di tali dati con altri non di "proprietà" dell'azienda, ma comunque accessibili da essa (ad esempio, statistiche fornite dall'ISTAT, dati sull'andamento della borsa); è allora necessaria una descrizione anche degli "schemi" di tali sorgenti.

Analisi dei Dati di Ingresso. La prima fase consiste in una analisi dei dati a disposizione sulla base dei requisiti dell'applicazione. Viene dapprima effettuata una analisi preliminare del patrimonio informativo, che consiste in una correlazione tra i requisiti e le sorgenti informative disponibili, al fine di selezionare le sorgenti necessarie (o loro porzioni) e individuare eventuali priorità tra schemi. Gli schemi selezionati vengono quindi trasformati in un modello concettuale dei dati di riferimento, per favorire la correlazione tra gli schemi e la loro integrazione. I diversi schemi vengono quindi analizzati al fine di identificare alcuni dei concetti su cui sarà basata l'analisi, nonché criteri per l'integrazione degli schemi.

Integrazione. Le descrizioni concettuali delle sorgenti informative vengono integrate, producendo lo schema concettuale globale del patrimonio informativo aziendale. Questo richiede l'esecuzione dei passi di una metodologia di integrazione, che assumiamo nota. L'integrazione viene guidata dai requisiti, nonché dai criteri di priorità identificati nella fase precedente. Va osservato che lo schema prodotto rappresenta una vista integrata del patrimonio informativo aziendale, ma possiede ancora caratteristiche tipiche delle basi di dati operazionali.

Progettazione del Data Warehouse. Questa fase prevede il progetto e la costruzione del data warehouse. La progettazione concettuale del data warehouse ristruttura lo schema integrato, al fine di introdurre quei concetti necessari per l'analisi che sono ancora assenti nello schema dei dati. La progettazione logica del data warehouse porta alla produzione dello schema logico corrispondente (tipicamente uno schema relazionale), ed elabora la documentazione prodotta al fine di identificare le trasformazioni necessarie alla materializzazione del data warehouse a partire dalle sorgenti informative selezionate.

Progettazione delle Basi di Dati Multidimensionali. Il data warehouse viene costruito per supportare tutte le esigenze aziendali di analisi. In generale, è opportuno estrarre porzioni dei dati del data warehouse, opportunamente aggregate, per realizzare le cosiddette basi di dati multidimensionali (o data mart). Viene

dapprima prodotto uno schema multidimensionale per ciascuna esigenza di analisi. In questa fase è opportuno utilizzare un modello logico di dati, in grado di rappresentare esplicitamente quei concetti caratteristici delle basi di dati multidimensionali in maniera indipendente dalle modalità di realizzazione; noi faremo riferimento al modello MultiDimensionale ($\mathcal{MD}$ [4, 5]). Quindi, per ciascuna base di dati multidimensionale viene prodotto il corrispondente schema nel modello dei dati del sistema OLAP a disposizione (relazionale o multidimensionale), nonché definiti i mapping per l'estrazione dei dati dal data warehouse.

2.2 I Modelli di Dati Usati nella Metodologia

Come si evince dal quadro metodologico, il progettista di un data warehouse ha la necessità di utilizzare una varietà di modelli di dati, in fasi diverse e con finalità diverse. Questo aspetto viene discusso nel presente paragrafo.

Innanzitutto, le varie sorgenti informative in ingresso al procedimento di progettazione sono da considerarsi preesistenti, siano esse di proprietà dell'azienda o no. Lo schema di tali sorgenti informative è generalmente disponibile, in un formato che dipende dalla modalità di realizzazione della sorgente stessa. In particolare, se la sorgente è strutturata, allora è descritta da uno schema logico di base di dati (relazionale o non) oppure mediante tracciati record. Altri casi vanno considerati individualmente; ad esempio, è ancora possibile pensare di descrivere dati "semistrutturati" accessibili mediante una interfaccia Web, almeno con riferimento alla loro porzione più regolare. Nei casi più fortunati, è disponibile la documentazione di progetto della sorgente informativa, in cui lo schema dei dati è descritto anche mediante un modello di dati concettuale.

Gran parte della metodologia opera su una descrizione dei dati a livello concettuale. Riteniamo infatti che la manipolazione di più schemi di dati risulti più efficace e più semplice in riferimento ad un modello concettuale, piuttosto che ad un modello logico. Noi faremo riferimento al modello Entità Relazione [3].

La progettazione del data warehouse viene effettuata prima al livello concettuale (utilizzando ancora il modello E-R) e poi al livello logico. In molti progetti di data warehousing è opportuno realizzare sia un unico data warehouse, che una o più base di dati multidimensionale. In questi casi, il data warehouse viene spesso opportunamente implementato utilizzando la tecnologia relazionale.

Infine, le basi di dati multidimensionali vengono generalmente implementate utilizzando un server OLAP. Anche in caso di adozione di un server OLAP relazionale, in cui è necessaria una rappresentazione mediante uno schema relazionale, riteniamo opportuno l'utilizzo di un modello di dati per basi di dati multidimensionale che sia indipendente da ogni implementazione. Come già detto, faremo riferimento al modello $\mathcal{MD}$ (descritto nel prossimo paragrafo), che è un modello logico di dati che si colloca ad un livello di astrazione maggiore che non il modello relazionale. In effetti, nella progettazione di basi di dati multidimensionali, è ragionevole considerare $\mathcal{MD}$ come il modello logico, e il modello relazionale come un modello "fisico," da utilizzare nella fase della progettazione che si occupa dell'allocazione dei dati e nell'ottimizzazione del loro accesso.

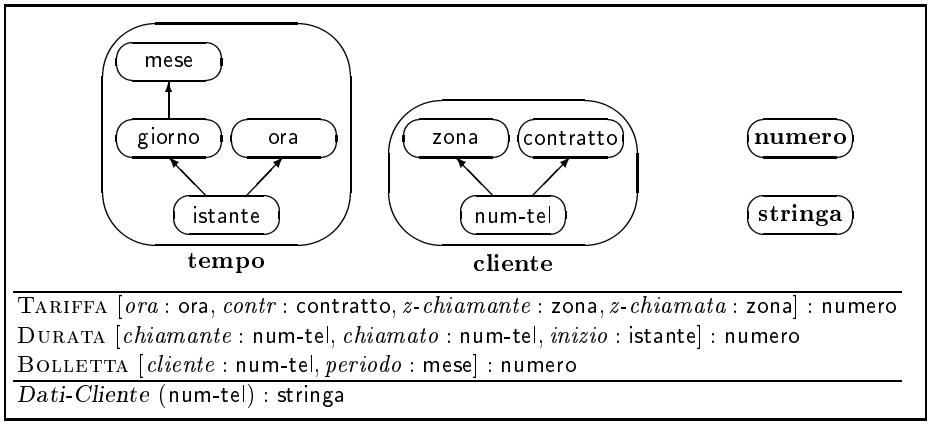

Figura2. Lo schema $\mathcal{MD}$ `TelCo`

2.3 Il Modello $\mathcal{MD}$

Il modello MultiDimensionale [4, 5] (abbreviato in $\mathcal{MD}$) si basa su due costrutti principali: dimensione e f-tabella. Le *dimensioni* sono categorie sintattiche che consentono di specificare "prospettive" secondo le quali vogliamo analizzare i dati. Ogni dimensione è organizzata in una gerarchia di *livelli*, che corrispondono essenzialmente a domini a livelli di granularità differenti. È possibile associare *descrizioni* ai livelli. In una dimensione, valori di livelli differenti sono correlati da una famiglia di *funzioni di roll-up*. Le *f-tabelle* sono funzioni che associano *misure* a *coordinate simboliche* (definite rispetto a una particolare combinazione di livelli): esse sono usate per rappresentare i dati fattuali.[2]

Consideriamo ad esempio una compagnia telefonica interessata all'analisi dei suoi dati operazionali. I dati relativi alle chiamate possono essere organizzate lungo le dimensioni **tempo** e **cliente**. Le corrispondenti gerarchie sono riportate in Figura 2. Le altre dimensioni sono usate per rappresentare valori numerici e stringhe (dimensioni atomiche **numero** e **stringa**). Il livello num-tel (numero telefonico) si aggrega su zona (l'area geografica dell'utenza telefonica, identificato da un codice) e contratto (caratterizzato da tariffe in ore differenti). Il dominio associato al livello instante contiene "timestamps" del tipo *5 Gen 97, 10:45:21*. Questo valore si aggrega in *10* al livello di ora (inteso nel senso di "ora del giorno") e in *5 Gen 97* a livello giorno. Diverse f-tabelle possono essere definite in questo contesto (come indicato nella stessa figura). TARIFFA rappresenta il costo per un minuto di conversazione tra un cliente in una *z(ona)-chiamante* (con un contratto di tipo *contr(atto)*) e un cliente in una *z(ona)-chiamata*, con inizio in una certa *ora*. La seconda f-tabella associa la DURATA in secondi ad ogni chiamata (fatta da un *chiamante* a un *chiamato* ad una certa ora). La f-tabella BOLLETTA è derivata e aggrega i costi per numero di telefono e mese. Infine,

[2] La "f" nel termine f-tabella ha un doppio significato: sta per "funzione," perché ogni f-tabella è in effetti una funzione da coordinate a misure; e per "fatto," perché rappresenta ciò che nei sistemi è comunemente memorizzato nelle "tabelle dei fatti" (fact-tables).

ora	*contr*	*z-chiamante*	*z-chiamata*	TARIFFA
6	*Family*	*06*	*02*	44
7	*Family*	*06*	*02*	72
8	*Family*	*06*	*02*	112
		...		...
6	*Pro*	*06*	*055*	80
7	*Pro*	*06*	*055*	80
8	*Pro*	*06*	*055*	135
		...		...

BOLLETTA	*Gen-97*	*Feb-97*	*Mar-97*
06-555-123	129*k*	231*k*	187*k*
06-555-456	429*k*	711*k*	664*k*
02-555-765	280*k*	365*k*	328*k*

Num-tel	*Dati-Cliente*
06-555-123	*Bruni*
06-555-456	*Dani*
02-555-765	*Sili*

Figura3. Una istanza dello schema `TelCo`

Dati-Cliente è una descrizione di livello che associa ad una utenza telefonica i dati anagrafici del cliente.

Una possibile istanza per lo schema `TelCo` viene riportato in Figura 3. Una coordinata simbolica sulla f-tabella TARIFFA è

$$[ora : 7, contr : Family, z\text{-}chiamante : 06, z\text{-}chiamata : 02].$$

L'istanza associa la misura 72 a questa entry. La descrizione *Dati-Cliente* associa la stringa *Bruni* al valore *06-555-123* del livello num-tel. Si osservi che due rappresentazioni grafiche diverse sono state usate, per una f-tabella, in Figura 3: una tabella tradizionale per TARIFFA e una matrice per BOLLETTA. Questo fatto suggerisce che è possibile implementare una f-tabella in diverse maniere.

3 Fasi per la Progettazione del Data Warehouse

La progettazione di un data warehouse richiede l'esecuzione delle prime tre macro-fasi tra le quattro previste dal quadro metodologico. Tra queste, le prime due sono prevalentemente composte da attività descritte da metodologie note in letteratura. In effetti, la progettazione di un data warehouse ha molte caratteristiche in comune con la progettazione di una base di dati integrata o federata, tranne che per le finalità, decisamente rivolte alla gestione di dati per l'analisi anziché ad attività transazionali. Nei Paragrafi 3.1 e 3.2 vengono illustrate brevemente le attività preliminari che portano essenzialmente alla definizione di uno schema concettuale integrato del patrimonio informativo aziendale. Il Paragrafo 3.3 è dedicato alla descrizione delle attività da svolgere per progettare il data warehouse (orientato all'analisi) a partire dallo schema concettuale integrato (che ha prevalentemente caratteristiche "transazionali").

3.1 Analisi dei Dati di Ingresso

La prima fase consiste in una attenta analisi dei dati a disposizione sulla base dei requisiti dell'applicazione e può essere suddivisa nelle seguenti sotto-fasi.

Selezione delle Sorgenti Informative. L'analisi preliminare del patrimonio informativo disponibile consiste in una sua correlazione con i requisiti. Alcune sorgenti informative (o loro porzioni) risulteranno irrilevanti ai fini dell'analisi, e verranno per questo motivo trascurate. Inoltre è possibile assegnare delle preferenze tra gli schemi; ad esempio, in riferimento alla rappresentazione di un certo concetto, sarà preferibile lo schema della sorgente informativa in cui tale concetto è gestito in modo più accurato e aggiornato.

Traduzione in un Modello Concettuale di Riferimento. Questa fase prevede la traduzione degli schemi delle sorgenti informative selezionate in un modello dei dati comune, di riferimento. Assumiamo di utilizzare come modello di dati comune un modello concettuale, ed in particolare il modello E-R. Riteniamo infatti che la correlazione di più schemi di dati sia più semplice ed efficace in riferimento a un modello concettuale piuttosto che ad un modello logico. Il prodotto è costituito da uno schema E-R per ciascuna sorgente informativa selezionata, corredato dalla opportuna documentazione di supporto (ad esempio, metadati da utilizzare per l'eventuale conversione di dati fattuali).

Analisi delle Sorgenti Informative. Questa fase, preliminare all'integrazione di schemi, ha lo scopo di individuare, nei vari schemi, concetti irrilevanti o significativi per l'analisi. Sono da considerarsi irrilevanti quei concetti presenti nello schema per motivi strettamente operativi; in questo caso non ci aspettiamo che l'analisi si concentri su tali dati. I corrispondenti concetti possono essere rimossi dai rispettivi schemi.

D'altra parte, è importante identificare i concetti candidati alla rappresentazione di "fatti," "misure" e "dimensioni" nelle specifiche attività di analisi orientata al supporto alle decisioni. In questo contesto, chiamiamo *fatti* i concetti dello schema E-R (entità, relazioni o attributi) sui quali il processo di analisi può essere centrato. Una *misura* è invece una proprietà atomica di un fatto che intendiamo analizzare (tipicamente un attributo numerico di un fatto o un conteggio delle sue istanze). Infine, una *dimensione* è un sottoinsieme dello schema E-R dato che descrive una prospettiva lungo la quale l'attività di analisi può essere effettuata. Questo aspetto sarà ripreso con maggior dettaglio nel Paragrafo 4.1. Inoltre, per concetti di questi tipi che risultano essere rappresentati in più schemi, è necessario identificare eventuali sovrapposizioni delle istanze corrispondenti e, in questo caso, delle priorità.

3.2 Integrazione

Le descrizioni concettuali delle sorgenti informative vengono integrate, producendo lo schema concettuale globale del patrimonio informativo aziendale. Una metodologia di integrazione di schemi concettuali è descritta ad esempio in [3]. L'integrazione può essere utilmente guidata dai requisiti, nonchè dai criteri di priorità

identificati nella fase precedente. (Ad esempio, la risoluzione di eventuali conflitti strutturali può essere guidata dal criterio che privilegia la rappresentazione di "fatti" come entità.) Viene inoltre prodotta la necessaria documentazione di supporto orientata alla integrazione dei dati.

3.3 Progettazione del Data Warehouse

Questa fase prevede il progetto e la costruzione del data warehouse ed è vantaggioso suddividerla in due passi distinti.

Progettazione Concettuale del Data Warehouse. Questa fase porta alla produzione dello schema concettuale del data warehouse a partire dallo schema integrato delle sorgenti informative a disposizione. Infatti, lo schema integrato prodotto dal passo precedente contiene i concetti necessari per le esigenze di analisi, ma con caratteristiche che sono ancora quelle di dati rappresentati per esigenze operative. È quindi necessario introdurre nello schema degli ulteriori concetti necessari per l'analisi, che sono spesso complementari a concetti già presenti nello schema. In questa fase devono essere introdotti, ad esempio, dati dimensionali ancora assenti (in particolare, concetti per la rappresentazione di dati storici) e dati aggregati. (Gli aspetti relativi all'introduzione di nuove dimensioni verrà ripreso con maggior dettaglio nel Paragrafo 4.1.) È possibile inoltre rimuovere concetti irrilevanti per l'analisi che non sono stati precedentemente identificati. Il prodotto di questa fase è lo schema concettuale del data warehouse, con la documentazione di supporto orientata alla estrazione e integrazione delle istanze dalle diverse sorgenti informative.

Progettazione Logica del Data Warehouse. A partire dallo schema concettuale del data warehouse, ne viene progettato il corrispondente schema logico (tipicamente, nel modello relazionale). Questa attività può essere svolta utilizzando metodologie note (vedi ancora [3]). I criteri di progettazione da utilizzare sono talvolta diversi da quelli tradizionali: infatti il data warehouse è spesso caratterizzato da un opportuno livello di denormalizzazione dei dati, con presenza di ridondanza, soprattutto per rappresentare dati derivati.

Deve essere inoltre prodotta la documentazione di supporto che descrive la sequenza di trasformazioni necessarie all'estrazione e all'integrazione dei dati dalle sorgenti informative al data warehouse. In particolare, questa deve descrivere sia i mapping da utilizzare per popolare inizialmente il data warehouse, che quelli necessari a propagare gli aggiornamenti.

4 Progettazione delle Basi di Dati Multidimensionali

Come osservato nell'Introduzione, l'analisi dei dati non avviene direttamente sul data warehouse, ma piuttosto nei data mart, ovvero in basi di dati multidimensionali derivate dal data warehouse. In effetti, i data mart sono spesso descritti in letteratura come dei data warehouse "dipartimentali," in quanto sono orientati al supporto di singole esigenze di analisi. Tuttavia, i dati del data mart

sono semplicemente estratti dal data warehouse già costruito, anziché estratti e integrati dalle singole sorgenti informative operazionali.

Questo paragrafo descrive la fase di progettazione di un data mart, a partire dallo schema concettuale del data warehouse ed in riferimento ad una singola esigenza di analisi. In particolare, la nostra metodologia sarà illustrata in riferimento allo schema concettuale del data warehouse `Rivendita`, illustrata in Figura 4, che possiamo supporre costruito utilizzando le fasi metodologiche descritte nel Paragrafo 3.

4.1 Progettazione Logica di una Base di Dati Multidimensionale

La fase di progettazione logica di una base di dati multidimensionale viene suddivisa in quattro passi: (i) identificazione di fatti e dimensioni; (ii) ristrutturazione dello schema E-R; (iii) derivazione di un "grafo dimensionale"; (iv) traduzione nel modello $\mathcal{MD}$. In effetti, i primi due passi non sono strettamente sequenziali, ma procedono, in molti casi, in parallelo: durante la ristrutturazione dello schema E-R, i fatti e le dimensioni selezionati possono essere raffinati e/o modificati. Successivamente, il processo procede sequenzialmente, poiché ogni passo richiede il completamento del passo precedente.

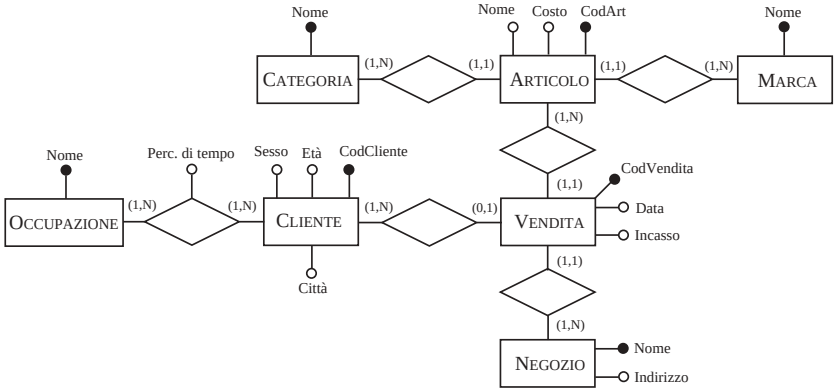

Figura4. Lo schema E-R che descrive il data warehouse `Rivendita`

Identificazione di Fatti e Dimensioni. La prima attività consiste in una analisi attenta dello schema E-R del data warehouse il cui scopo è la selezione di fatti, misure e dimensioni di interesse per la specifica attività di analisi gestionale orientata al supporto alle decisioni. Questa attività riprende una attività analoga svolta nel contesto dell'analisi delle sorgenti informative (Paragrafo 3.1), ma contestualizzandola all'esigenza in esame. Ad esempio, è possibile identificare che un concetto è dimensione per l'analisi di un altro concetto solo dopo che l'integrazione degli schemi è stata svolta. D'altra parte, un concetto considerato dimensione nel data warehouse potrebbe risultare irrilevante ai fini della specifica esigenza di analisi in esame.

Consideriamo lo schema E-R del nostro esempio riportato in Figura 4: potremmo essere interessati, da un lato, all'identificazione degli andamenti delle vendite (e dei rispettivi incassi), dall'altro, all'analisi della variazione temporale dei costi di produzione degli articoli in vendita. In questo caso i fatti sono individuabili nell'entità VENDITA e nell'attributo *Costo* dell'entità ARTICOLO. Le misure del primo fatto sono il volume delle vendite (conteggio delle istanze dell'entità corrispondente) e gli incassi relativi (attributo *Incasso*). L'unica misura per il secondo fatto è il valore del costo stesso. Si osservi che, in alcuni casi, un fatto ha diversi aspetti da analizzare, mentre in altri, la misura di un fatto coincide con il fatto stesso.

Per individuare una dimensione ci possiamo basare sulla osservazione che l'analisi di un fatto si effettua consolidando (cioè aggregando) i dati lungo una o più dimensioni [8]. Quindi, una dimensione può essere identificata navigando lo schema a partire dai fatti e includendo concetti che forniscono una maniera per raggruppare istanze di fatti (ad esempio, relazioni uno-a-molti, oppure i cosiddetti *attributi categorici*, come età e sesso). Consideriamo, ad esempio, l'entità-fatto VENDITA. Ogni vendita è correlata al corrispondente articolo venduto e ogni articolo è correlato alla rispettiva categoria e marca. Ne consegue che le vendite possono essere esaminate sulla base della tipologia del prodotto a differenti livelli di aggregazione (singolo prodotto, categoria di prodotto, marca). Quindi, una possibile dimensione per l'analisi delle vendite è la tipologia del prodotto venduto, dimensione che contiene le entità ARTICOLO, MARCA e CATEGORIA e le relazioni tra esse. Osserviamo inoltre che per alcune vendite conosciamo il relativo cliente; i clienti possono essere raggruppati per età, sesso e città di residenza (in accordo ai corrispondenti attributi categorici) e occupazione. Quindi la tipologia del cliente è un'altra dimensione per l'analisi delle vendite. Questa dimensione include le entità CLIENTE e OCCUPAZIONE e le relazioni tra di esse. Sulla base di considerazioni simili, possiamo concludere che il luogo delle vendite è un'altra possibile dimensione per la loro analisi: questa dimensione contiene (per il momento) la sola entità NEGOZIO. Infine, è facile identificare una dimensione temporale per l'analisi delle vendite (attributo *Data* dell'entità VENDITA): si tratta in effetti di una dimensione fondamentale, usata in pratica in tutte le analisi multidimensionali.

Ristrutturazione dello Schema E-R. Questa attività consiste in una riorganizzazione dello schema E-R originale del data warehouse il cui scopo è rappresentare fatti e dimensioni in maniera più esplicita ed efficace per lo scopo prefissato. L'obiettivo di questo passo è la produzione di uno schema E-R che possa essere tradotto direttamente nel modello $\mathcal{MD}$. Crediamo che sia conveniente effettuare questa attività nell'ambito del modello E-R poiché in questa maniera le corrispondenti trasformazioni tra le basi di dati operazionali e le basi di dati multidimensionali possono essere derivate più facilmente. La ristrutturazione può essere divisa in alcune attività di base, come descritto nel seguito.

Rappresentazione di Fatti mediante Entità. In genere, i fatti corrispondono ad entità dello schema E-R iniziale ma, come abbiamo visto, possono essere descritti

da attributi o relazioni. In questi casi, devono essere trasformati in entità (sulla base delle note trasformazioni che preservano il contenuto informativo [3]) poiché i fatti diventano di interesse primario nell'analisi multidimensionale. Come vedremo, questa trasformazione semplifica, tra l'altro, i passi che seguono.

Ad esempio, nella nostra applicazione, i costi di produzione sono rappresentati per mezzo di un attributo. Questo attributo può essere facilmente trasformato in un'entità COSTO ARTICOLO come mostrato in Figura 5, aggiungendo una relazione uno-a-uno tra la nuova entità e l'entità ARTICOLO. Le istanze di questa nuova entità sono identificate esternamente dall'articolo corrispondente.

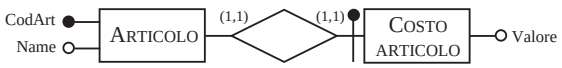

Figura5. Una ristrutturazione dell'entità ARTICOLO dello schema E-R in Figura 4

Individuazione di Nuove Dimensioni. Può accadere che, per certi fatti dello schema E-R, esistono dimensioni di interesse per l'analisi che non sono presenti nello schema (e quindi, sulla base delle nostre ipotesi, non sono rappresentate nelle basi di dati operazionali) ma possono essere dedotte o da basi di dati esterne o da meta-informazioni associate con le sorgenti informative. Ad esempio, potremmo accedere a informazioni circa la validità temporale delle istanze di un fatto o all'origine geografica di certi dati. Queste informazioni si possono tradurre in dimensioni che però vanno rappresentate esplicitamente nel nostro schema.

Consideriamo il costo degli articoli nella nostra base di dati `Rivendita`. È ragionevole che nelle transazioni operazionali siamo solo interessati al costo corrente di un articolo e quindi non disponiamo in linea di informazioni storiche sui costi. Supponiamo comunque di sapere gli istanti in cui le operazioni di aggiornamento vengono effettuate e che i costi cambiano, in media, una volta al mese. Dato che l'analisi effettiva dei costi può essere effettuata solo se siamo in grado di confrontare costi in periodi differenti, le informazioni temporale vanno aggiunte esplicitamente. In base alle meta-informazioni di cui disponiamo, questo può essere fatto ristrutturando l'entità COSTO ARTICOLO come descritto in Figura 6. Da un punto di vista pratico, i dati temporali possono essere ottenuti per mezzo di aggiornamenti incrementali che aggiungono, ogni mese, una nuova istanza dell'entità COSTO ARTICOLO, sulla base del valore corrente dell'attributo *Costo* nella base di dati originale.

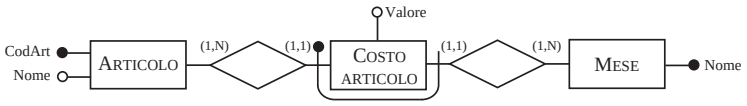

Figura6. Una ristrutturazione dell'entità COSTO ARTICOLO in Figura 5

Raffinamento dei Livelli di ogni Dimensione. All'interno di una dimensione, dobbiamo individuare e rappresentare esplicitamente i vari livelli di aggregazione che sono di interesse nell'analisi dei fatti (ad esempio, categoria e marca degli articoli). In particolare, i livelli vanno distinti dai concetti che sono solo descrittivi

ma non possono essere usati nell'attività di analisi multidimensionale, in quanto non permettono di effettuare aggregazioni (ad esempio, indirizzo e numero di telefono di un negozio). In pratica, questo passo richiede di effettuare una delle seguenti operazioni elementari: (1) sostituzione di relazioni molti-a-molti, (2) ristrutturazione di concetti (entità o attributi) per rappresentare nuovi livelli di interesse, (3) selezione di un identificatore per ogni entità che rappresenta un livello, (4) rimozione di concetti irrilevanti.

Vediamo in che maniera si applicano queste operazioni facendo riferimento alla nostra applicazione. Consideriamo la dimensione **cliente**: all'interno di questa dimensione possiamo aggregare i clienti in base all'età, al sesso e alla città di residenza (attraverso i corrispondenti attributi dell'entità CLIENTE). Ora se vogliamo aggregare i clienti anche rispetto alla loro occupazione, non possiamo usare direttamente la corrispondente entità poiché, in base alla relazione molti-a-molti tra CLIENTE e OCCUPAZIONE, ogni cliente ha in generale diverse occupazioni. Possiamo però, in base ai dati disponibili, rimpiazzare questa entità con una nuova entità OCCUPAZIONE PRINCIPALE, descrivente l'occupazione di un cliente nella maggior parte del tempo, in modo che la relazione si trasforma da molti-a-molti in uno-a-molti (vedi Figura 7). Consideriamo ora la dimensione **luogo** che contiene solo l'entità NEGOZIO. Potremmo essere interessati nell'aggregare i negozi sulla base della città e dell'area geografica (si osservi che queste informazioni possono essere derivate dall'attributo *Indirizzo* e da conoscenza "built-in"). Questa situazione può essere resa esplicita attraverso una ristrutturazione che introduce le entità CITTÀ e ZONA come mostrato in Figura 7. Per le nuove entità è importante scegliere un identificatore semplice (possibilmente naturale se esiste). Consideriamo infine la dimensione **tempo**, assumendo che l'altra dimensione **prodotto** non richieda ristrutturazioni. Potremmo voler aggregare le vendite su base giornaliera, mensile, trimestrale, annuale e per periodi speciali (ad esempio, Natale, Pasqua, Apertura scuole). Questo può essere reso esplicito, ancora una volta in base a conoscenza "built-in", mediante una ristrutturazione che introduce nuove entità e relazioni come mostrato in Figura 7.

Quando tutte le dimensioni sono state esaminate, vanno rimossi tutti i concetti dello schema (entità, attributi e relazioni) che non sono utili alla specifica esigenza di analisi (ad esempio, livelli di aggregazione non interessanti).

Lo schema E-R della nostra applicazione che si ottiene dopo la fase di ristrutturazione è riportato in Figura 7. Si osservi che lo schema è stato annotato con fatti e dimensioni. Si osservi inoltre che le dimensioni non contengono gli attributi descrittivi (ad esempio, l'attributo *Indirizzo* dell'entità NEGOZIO).

Derivazione di un Grafo Dimensionale. Partendo dallo schema E-R ristrutturato, possiamo ora derivare un grafo speciale che chiameremo *dimensionale*. Un grafo dimensionale rappresenta, in maniera succinta e facilmente comprensibile, fatti e dimensioni dello schema E-R di partenza. In particolare, ogni nodo del grafo corrisponde a un concetto specifico (entità o attributo) e rappresenta un dominio (insieme di valori) come segue: se il nodo corrisponde ad un'entità, esso rappresenta il dominio delle chiavi dell'entità; se il nodo corrisponde ad

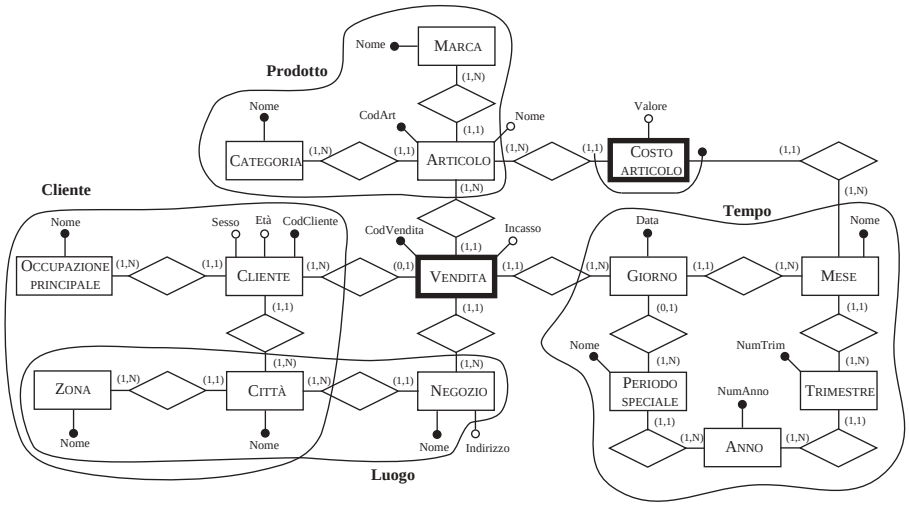

Figura7. Lo schema E-R di Figura 4 dopo la fase di ristrutturazione

un attributo, esso rappresenta il dominio dell'attributo. Un arco tra due nodi rappresenta una funzione tra i corrispondenti domini (l'arco è tratteggiato se la funzione è parziale). La Figura 8 riporta il grafo dimensionale che si ottiene dallo schema E-R di Figura 7. In base a quanto detto, in questo grafo il nodo ARTICOLO rappresenta il dominio dell'attributo *CodArt* e il nodo MESE rappresenta il dominio dell'attributo *Nome* della corrispondente entità; per contro, il nodo INCASSO rappresenta il dominio dell'attributo *Incasso* dell'entità VENDITA. È facile comprendere che questo grafo può essere ottenuto automaticamente e che possiede lo stesso contenuto informativo dello schema E-R originale. Si osservi inoltre che le dimensioni diventano sottografi del grafo dimensionale. In un grafo dimensionale è possibile distinguere quattro tipi di nodi: *nodi fatto*, denotati da margini in grassetto (originano da entità fatto); *nodi livello*: sono quelli contenuti in una dimensione; *nodi descrittivi*: sono nodi non contenuti in dimensioni e hanno un arco entrante che esce da un nodo livello (originano da attributi descrittivi); *nodi misura*: sono nodi non contenuti in dimensioni e hanno un arco entrante che esce da un nodo fatto (essi originano da misure).

Traduzione nel Modello $\mathcal{MD}$. Da un grafo dimensionale è possibile ottenere direttamente le $\mathcal{MD}$-dimensioni: ce ne sarà una per ogni dimensione del grafo e, per ognuna di esse, avremo un $\mathcal{MD}$-livello per ogni nodo e una funzione di roll-up per ogni arco del corrispondente sottografo. Tali sottografi denotano anche l'ordinamento parziale sugli $\mathcal{MD}$-livelli. Quello che rimane da fare è definire le dimensioni atomiche necessarie per rappresentare nodi misura e nodi descrittivi. Nel nostro caso, possiamo definire una dimensione **numerico** per gli incassi e i costi; e una dimensione **string** per i nomi degli articoli e gli indirizzi dei negozi. Infine, dobbiamo introdurre una $\mathcal{MD}$-descrizione per tutti i nodi descrittivi: nel

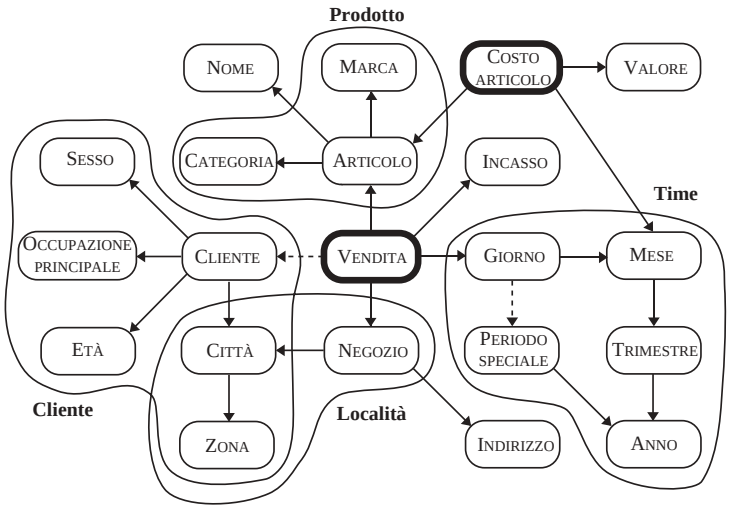

Figura8. Il grafo dimensionale ottenuto dallo schema in Figura 7

nostro caso avremo la descrizione *Nome* per il livello articolo e la descrizione *Indirizzo* per il livello negozio.

A questo punto, le f-tabelle possono essere definite come segue. Per ogni nodo fatto del grafo dimensionale, selezioniamo una combinazione di livelli appartenenti alle dimensioni "associate" al fatto, quelle cioè per le quali esiste un arco uscente dal nodo fatto che incide su di esse (in particolare sul loro livello base). Per ogni dimensione si possono selezionare più livelli per ciascuna dimensione e non è necessario selezionare un livello per ogni dimensione associata al fatto. Dobbiamo quindi definire un mapping θ, basato eventualmente su aggregazioni, che descriva il risultato della f-tabella. Questo mapping può essere (i) il conteggio di una collezione di fatti, oppure (ii) una espressione su una misura. Le istanze delle f-tabelle sono poi definite di conseguenza [5].

Nell'applicazione `Rivendita`, avevamo identificato tre misure: (1) il numero di articoli venduti, (2) gli incassi e (3) il costo degli articoli. Le prime due misure vanno descritte giornalmente per ogni articolo e negozio, la terza va descritta su base mensile. Queste misure possono essere descritte dalle seguenti f-tabelle.

1. VENDITA[*periodo* : giorno, *prodotto* : articolo, *luogo* : negozio] : numerico, definita sul fatto VENDITA dal mapping **count**(VENDITA);
2. INCASSOTOT[*periodo* : giorno, *prodotto* : articolo, *luogo* : negozio] : numerico, definita sul fatto VENDITA dal mapping **sum**(INCASSO(VENDITA));
3. COSTOARTICOLO[*periodo* : mese, *prodotto* : articolo] : numerico, definita sul fatto COSTO ARTICOLO dal mapping VALORE(COSTO ARTICOLO).

Potremmo essere interessati anche a dati parzialmente aggregati. Ad esempio, l'analisi degli acquisti dei clienti per età, categoria di prodotto e anno può essere effettuata con la seguente f-tabella:

VENDITEPERETA[*età* : età, *prodotti* : categoria, *periodo* : anno] : numerico,

definita sul fatto VENDITA dal mapping **sum**(INCASSO(VENDITA)).

4.2 Implementazione di una Base di Dati Multidimensionale

Una base di dati multidimensionale può essere implementata mediante un sistema OLAP. Viene di seguito brevemente illustrata l'implementazione mediante tabelle relazionali. (Una descrizione più dettagliata dell'implementazione di basi di dati multidimensionali mediante sistemi OLAP relazionali e matrici multidimensionali è presentata in [5].)

La rappresentazione naturale di una base di dati multidimensionale nel modello relazionale consiste in uno "schema a stella" (*star scheme* [13, 14]) contenente "tabelle fatto" e "tabelle dimensione". Le prime sono normalizzate mentre le seconde possono essere denormalizzate. Facciamo ora vedere come una base di dati $\mathcal{MD}$ può essere memorizzata in uno schema relazionale a stella. L'approccio descritto può essere adattato a varianti del modello a stella (per esempio, agli "schemi a fiocco di neve" o snowflake).

Lo schema stella rappresentante una base di dati $\mathcal{MD}$ può essere costruito come segue. Abbiamo: (1) uno schema di relazione R_d per ogni dimensione non-atomica d, e (2) uno schema di relazione R_f per ogni f-tabella f. Le dimensione atomiche non vanno rappresentate esplicitamente dato che generalmente corrispondono a domini di base. Ad esempio, la rappresentazione a schema stella della base di dati $\mathcal{MD}$ `Rivendita` è mostrata in Figura 9 (l'istanza è solo accennata).

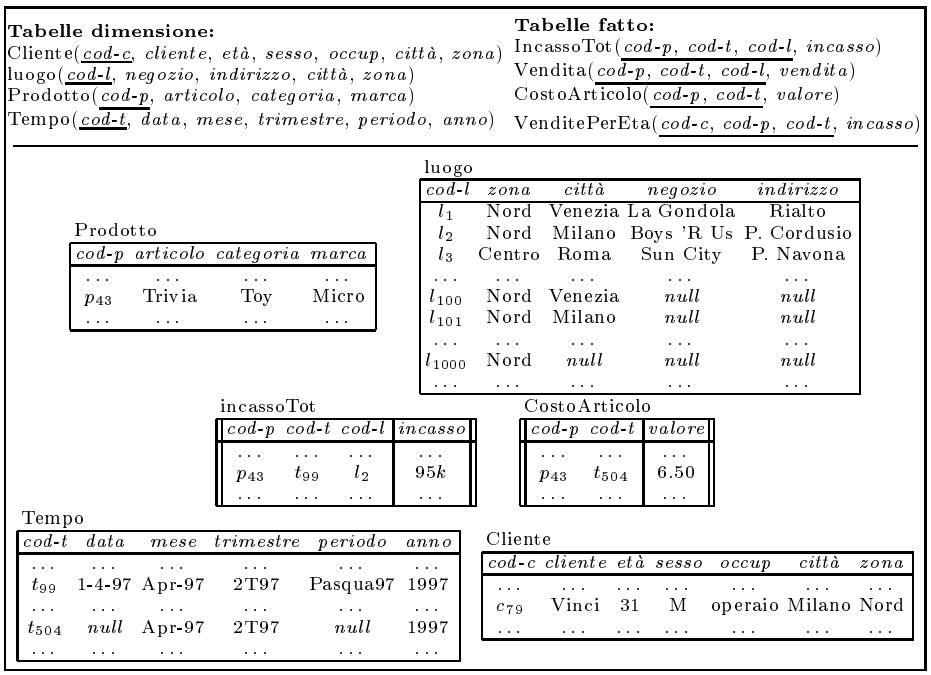

Figura9. Lo schema a stella della base di dati multidimensionale `Rivendita`

Riferimenti bibliografici

1. S. Agarwal et al. On the computation of multidimensional aggregates. In *Twenty-second Int. Conf. on Very Large Data Bases, Bombay*, pages 506–521, 1996.
2. R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *Thirteenth Int. Conf. on Data Engineering*, pages 232–243, 1997.
3. C. Batini, S. Ceri, and S. Navathe. *Conceptual Database Design.* Benjamin/Cummings, 1992.
4. L. Cabibbo and R. Torlone. Querying multidimensional databases. In *Sixth Int. Workshop on Database Programming Languages (DBPL'97), Springer-Verlag*, 1997.
5. L. Cabibbo and R. Torlone. A logical approach to multidimensional databases. In *Int. Conf. on Extending Data Base Technology (EDBT'98), Springer-Verlag*, 1998.
6. D. Chatziantoniou and K. Ross. Querying multiple features of groups in relational databases. In *Twenty-second Int. Conf. on Very Large Data Bases, Bombay*, pages 295–306, 1996.
7. S. Chaudhuri and U. Dayal. An overview of Data Warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74, March 1997.
8. E.F. Codd, S.B. Codd, and C.T. Salley. Providing OLAP (On Line Analytical Processing) to user-analysts: An IT mandate. Arbor Software White Paper, *http://www.arborsoft.com.*
9. G. Colliat. OLAP, relational, and multidimensional database systems. *ACM SIGMOD Record*, 25(3):64–69, September 1996.
10. M. Golfarelli, D. Maio, and S. Rizzi. Conceptual design of data warehouses from E/R schemes. In *31st Hawaii Intl. Conf. on System Sciences*, 1998.
11. J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data Cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In *Twelfth IEEE International Conference on Data Engineering, Vienna*, pages 152–159, 1996.
12. M. Gyssens and L.V.S. Lakshmanan. A foundation for multi-dimensional databases. In *Twenty-third Int. Conf. on Very Large Data Bases, Athens*, pages 106–115, 1997.
13. W.H. Inmon. *Building the Data Warehouse.* John Wiley & Sons, second edition, 1996.
14. R. Kimball. *The Data Warehouse toolkit.* John Wiley & Sons, 1996.
15. A. O. Mendelzon. Data warehousing and OLAP: a research-oriented bibliography. *http://www.cs.toronto.edu/∼mendel/dwbib.html.*
16. N. Pendse and R. Creeth. The OLAP Report. *http://www.olapreport.com.*
17. S. Rao, A. Badia, and D. Van Gucht. Providing better support for a class of decision support queries. In *ACM SIGMOD International Conf. on Management of Data*, pages 217–227, 1996.
18. A. Shoshani. OLAP and statistical databases: Similarities and differences. In *Sixteenth ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems*, pages 185–196, 1997.
19. Stanford Technology Group, Inc. Designing the data warehouse on relational databases, 1995. Unpublished manuscript.
20. Red Brick Systems. Decision-makers, business data, and RISQL, 1995. White Paper, *http://www.redbrick.com.*