POLITECNICO DI MILANO

Facoltà di Ingegneria

Dipartimento di Ingegneria Gestionale

STUDIO DELLA QUALITÀ IN UN DWH DI UNA MULTINAZIONALE FARMACEUTICA TRAMITE METODOLOGIA DWQ E REALIZZAZIONE DI UN DATA MART PER LA QUALITÀ

Relatore: Ing. Cinzia CAPPIELLO

Elaborato di Laurea di:

Simone RIGO Matr. n. 680232

Anno Accademico 2010-2011

Indice dei contenuti

Indice delle figure

Indice delle tabelle

Abstract

Un Data Warehouse (DWH) è un insieme di tecnologie che ha lo scopo di facilitare le persone di business nel prendere le migliori decisioni possibili il più velocemente possibile. Ci si aspetta, quindi, che il DWH presenti le informazioni corrette al momento e nel luogo opportuno e al giusto costo, in maniera tale da poter prendere le decisioni di business corrette.

L'esperienza ci insegna che i tradizionali sistemi di processi transazionali on-line (denominati OLTP) non sono appropriati per il supporto alle decisioni e che la sola rete, nonostante i notevoli progressi di velocità di trasmissione dei dati, non è in grado di risolvere il problema dell'accessibilità alle informazioni. Il DWH è diventato, quindi, un' importante strategia per integrare informazioni eterogenee provenienti da più fonti, in modo tale da poter realizzare dei processi analitici direttamente on-line (denominati OLAP).

Il fenomeno del DWH è una conseguenza dell'osservazione di W. Inmon e F. Codd dei primi anni '90 secondo cui i livelli operazionali OLTP e OLAP non possono coesistere efficientemente nello stesso database per due principali ragioni, che richiedono entrambe dei compromessi sulla qualità del dato:

- **Le caratteristiche dei dati**: i database OLTP mantengono dati correnti con un grande livello di dettaglio per il loro immediato utilizzo, mentre i sistemi OLAP lavorano con dati aggregati e spesso storici, che coprono un periodo temporale ben più ampio rispetto ai dati correnti; l'utilizzo contemporaneo di questi dati comporta complessi compromessi dovuti sia alla granularità dei dati stessi sia alla necessità di diversa profondità storica delle informazioni.
- **Le caratteristiche delle transazioni**: i sistemi OLTP enfatizzano l'efficienza di brevi transazioni che vadano ad aggiornare una piccola parte del database mentre i sistemi OLAP richiedono interrogazioni complesse che coprono una buona parte del database; mischiare questo tipo di transazioni comporta problemi nel controllo della concorrenza all'accesso dei dati.

Il DWH, quindi, prende specifici dati che sono di interesse per un determinato gruppo di clienti, in modo tale da poter accedere a questi dati in maniera veloce, economica e più efficiente. Le domande che dobbiamo affrontare, però, sono due: come posso riconciliare il flusso di dati in ingresso da un insieme di fonti dati eterogenee? E come possiamo personalizzare il derivante immagazzinamento delle informazioni per le specifiche applicazioni OLAP?

I compromessi che guidano le decisioni di come debba essere realizzato il DWH, per quanto riguarda le due domande precedenti, cambiano continuamente a seconda dei bisogni del business, quindi la gestione del supporto e dei cambiamenti del design di un DWH sono molto importanti, per non portare il lavoro in un vicolo cieco.

I venditori concordano col fatto che i DWH non possono essere dei prodotti a se stanti, ma devono essere progettati e ottimizzati ponendo molta attenzione alle necessità del cliente. Le tecniche di progettazione dei database tradizionali non possono essere applicate poiché non possono risolvere problemi specifici di un DWH come la selezione di un determinato subset di dati, aggregazione e profondità storica dei dati e la gestione dei controlli di ridondanza delle informazioni.

Dal momento che la grande varietà di prodotti e strategie dei venditori impediscono di trovare una soluzione a basso-livello per questi problemi di progettazione a costi accettabili, solo un arricchimento dei servizi di metadati che congiungono implementazioni eterogenee costituisce una soluzione promettente. Questo è quello che si propone di fare la ricerca del data warehouse quality (DWQ) ed è quello che andremo ad analizzare all'interno di questo trattato.

Il DWQ è un progetto cooperativo all'interno del programma ESPRIT della comunità europea. Il suo scopo è quello di stabilire dei punti fermi riguardo alla qualità dei dati nel DWH legando modelli semantici di architetture DWH a modelli espliciti di controllo di qualità del dato (DWQ – Foundations of Data Warehouse Quality).

Nel prossimo capitolo approfondiremo la metodologia del DWQ nello sviluppare fondamenti semantici che permettano ai progettisti di DWH di guidare le loro scelte verso modelli più profondi, strutture dati più ricche e rigorose, e tecniche di implementazione di fattori di qualità sistematici, così da migliorare il design, le operazioni e soprattutto l'evoluzione delle applicazioni del DWH.

Verrà poi preso in considerazione un DWH esistente di un'azienda farmaceutica multinazionale, per analizzare se e come queste metodologie vengono utilizzate e, laddove ci siano delle lacune, in quale modo intervenire per colmarle.

1. Architettura di un Data warehouse; dalla concezione standard alla concezione estesa

L'architettura tradizionale di un DWH, descritta sia nella ricerca che nei giornali di vendita commerciale, è la seguente

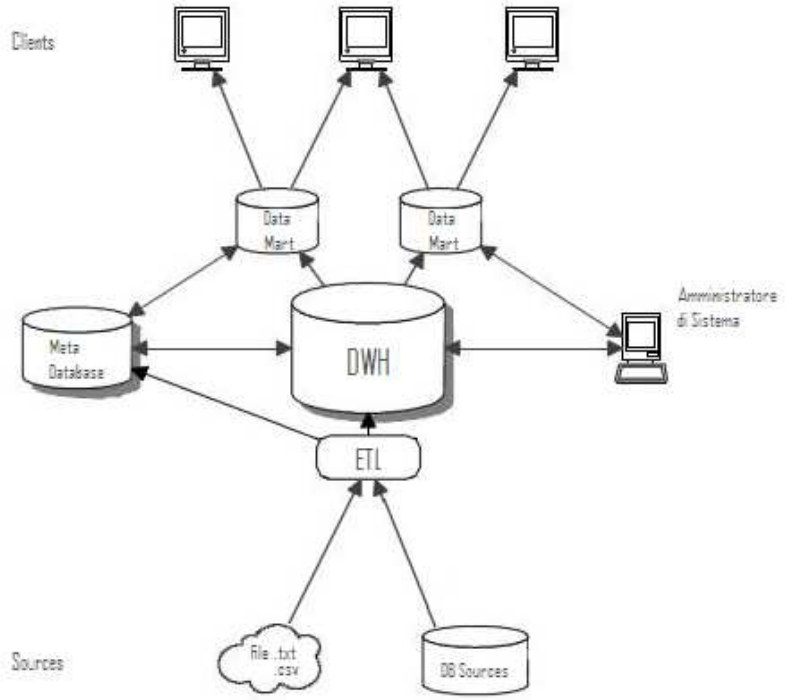


Figura 1 - Architettura di un DWH (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Fisicamente, un data warehouse è formato da più basi di dati (DB) che sono sia fonti da cui si estrapolano i dati, sia schemi in cui vengono copiati ed elaborati, sia viste materializzate per le applicazioni finali; inoltre il DWH è composto anche da agenti che trasportano dati da un DB all'altro (denomitati ETL, il cui acronimo significa Extract, Transform, Load) .

Come si evince dalla figura 1, gli ETL hanno il compito di caricare le informazioni che arrivano dalle fonti dati (che possono essere semplici file di testo o .csv, oppure query che leggono dati direttamente su schemi in lettura di diversi DB, come quello del sistema CRM o del sistema ERP dell'azienda stessa), dopodiché si fanno carico di integrare le informazioni caricate, aggregarle a seconda dell'utilizzo che se ne deve fare e risolvere gli eventuali problemi di duplicazione e dati non consistenti. Questi dati "puliti" vengono quindi immagazzinati nel DWH. Le tabelle e le viste del DWH sono solitamente poco aggregate; per adattarsi meglio alle esigenze dei gruppi di utenti, vengono creati dei data mart di secondo livello che vengono letti direttamente dalle applicazioni di business intelligence.

Lo scopo di questo studio è quello di definire uno schema di meta database che possa catturare e collegare tutti gli aspetti rilevanti di un'architettura di un DWH e dei problemi di qualità dei dati. Questo processo verrà descritto in diversi passi; per prima cosa dovremo aggiungere un livello denominato "concettuale" nell'architettura del DWH, che servirà ad eliminare alcuni dei problemi di qualità propri dell'architettura tradizionale. In secondo luogo elaboreremo il meta modello esteso e vedremo come si può realizzare in un repository. Infine illustreremo l'applicazione dei principi relativi ai repository con la descrizione di un modello specifico più dettagliato.

1.1. Aggiunta del livello concettuale in un Data Warehouse

Come abbiamo visto fin ora, l'architettura di un DWH viene considerata come un flusso di informazioni passo a passo dalle fonti dati fino, attraverso determinate aggregazioni, agli utenti finali. L'osservazione principale del nuovo approccio è che l'architettura in figura 1 copre solo parzialmente i compiti di un DWH, ed è inoltre incapace di esprimere un gran numero di importanti problemi di qualità e di strategie di management.

L'argomento principale che vogliamo introdurre è la necessità di un modello concettuale dell'azienda. Per spiegarlo prendiamo in considerazione la figura sottostante:

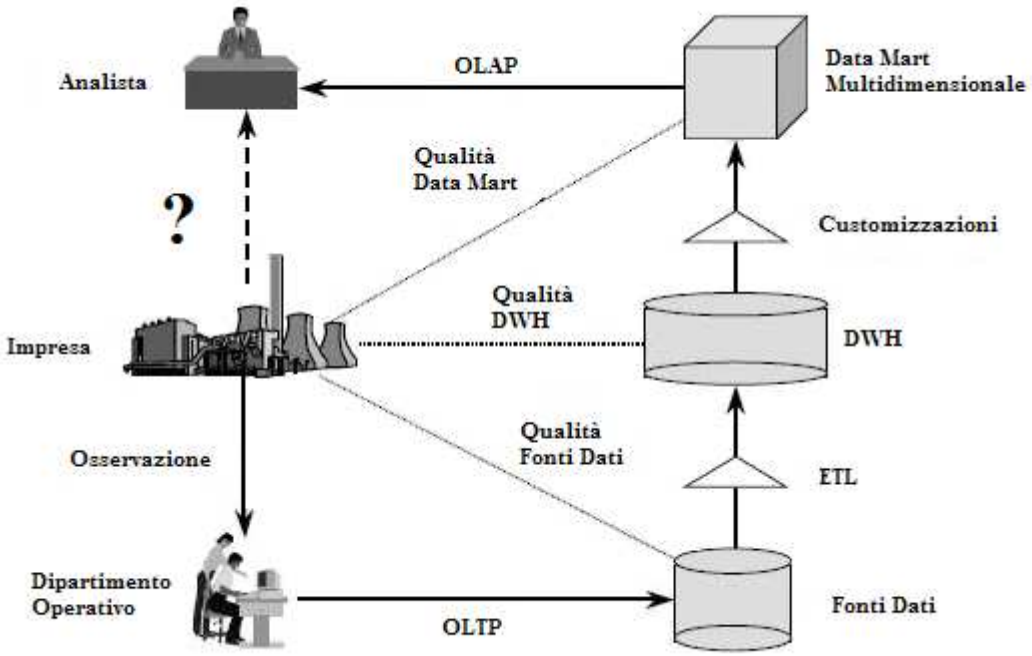


Figura 2 - DWH in un contesto di impresa (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

In questa figura, il flusso delle informazioni descritto nel paragrafo precedente è stilizzato nella colonna a destra, mentre il processo di creazione e utilizzo delle informazioni è mostrato a sinistra. Supponiamo ora che un'analista voglia conoscere qualcosa del business (indicato con il punto di domanda nella figura 2) e che non abbia sufficientemente tempo per osservare direttamente il business, ma debba fare affidamento solo dalle informazioni ottenute dal dipartimento operazionale e documentate come effetto collaterale del sistema OLTP.

Questo modo di estrarre le informazioni implica già di per sé uno svantaggio che deve essere compensato quando si selezionano i dati dal sistema OLTP per il caricamento e la pulizia di dati in un DWH, dove sono poi pre-processati ed aggregati nei data mart per i programmi di analisi. Considerando il lungo percorso che viene fatto dai dati, risulta ovvio che anche l'ultimo passo, la formulazione di query e l'interpretazione delle risposte concettualmente adeguate rappresentano uno dei problemi maggiori per l'analista.

La letteratura tradizionale del DWH, però, copre solo 2 dei 5 passi indicati nella Figura2 (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999). Per questo motivo, non si ha risposta a domande del tipo

"come mai il mio dipartimento operazionale utilizza così tanti soldi per la qualità del dato e la qualità del mio DWH è ancora così bassa?" oppure "qual è lo sforzo richiesto per analizzare il problema X per il quale il DWH, al momento attuale, non offre informazioni?". Una risposta adeguata a queste domande richiede un modello esplicito delle relazioni concettuali che intercorrono tra il modello dell'impresa, le informazioni acquisite dai dipartimenti OLTP e i clienti OLAP che hanno il compito di prendere le decisioni.

In precedenza abbiamo già detto che un DWH è il maggiore investimento che viene fatto per un determinato proposito di business. Non introduciamo, quindi, il modello di impresa come una parte minore del sistema, ma vogliamo che tutti gli altri modelli siano definiti come viste che si basano sul modello di impresa. In maniera del tutto sorprendente, possiamo dire che anche gli schemi delle fonti dati definiscono viste sul modello di impresa, non viceversa come indicato nella Figura 1.

1.2. Un modello di repository per l'architettura estesa del DWH

Introducendo una prospettiva di business esplicita, come indicato in Figura 2, i caricamenti e le trasformazioni attuate dagli ETL nella letteratura classica del DWH possono, quindi, essere tutte controllate in termini di interpretazione, consistenza o completezza rispetto al modello di impresa. Allo stesso tempo, le trasformazioni logiche devono essere implementate in maniera sicura ed efficiente dal data storage fisico e dagli ETL stessi; questo è la terza prospettiva nel nostro modello. È chiaro che gli aspetti di qualità della prospettiva fisica richiedono parametri completamente differente rispetto, ad esempio, a quelli della prospettiva concettuale; qui si richiederà tecniche di ottimizzazione delle query e della base dati.

Di conseguenza, il Meta Framework del DWH che viene proposto separa chiaramente le 3 prospettive, come mostrato in Figura3: una prospettiva concettuale dell'impresa, una prospettiva logica di modellazione dei dati e un prospettiva fisica del flusso dei dati (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999).

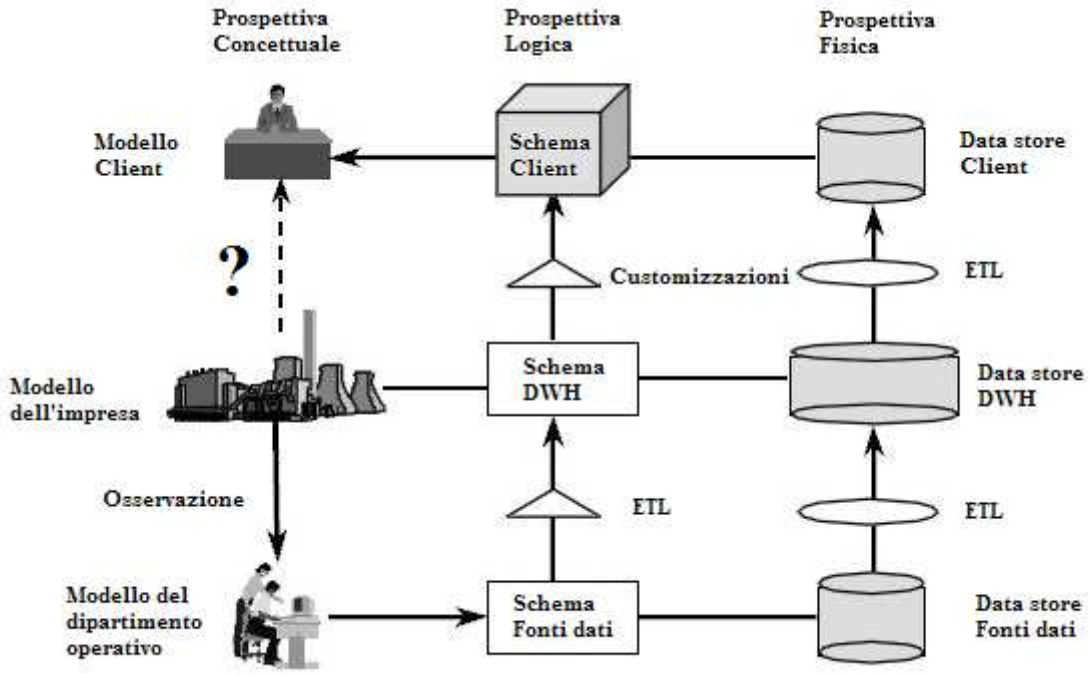


Figura 3 - Il Meta Data Framework proposto per il DWH (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Analizziamo ora in dettaglio ogni singola prospettiva:

1.2.1. Prospettiva concettuale

La prospettiva concettuale offre un modello di business dei sistemi informativi di un'impresa. Il ruolo centrale è dato al modello di impresa, che da una panoramica degli oggetti concettuali di un impresa. I modelli dei sistemi informativi del client e delle fonti dati vengono visti con l'ottica del modello di impresa, cioè i contenuti di questi modelli vengono descritti con terminologia del modello di impresa. Uno degli obiettivi della prospettiva concettuale è quello di astrarre un modello di informazioni indipendente dall'organizzazione fisica dei dati, in maniera tale da poter analizzare le relazioni che intercorrono tra i concetti per mezzo di programmi di business intelligence (BI). Dalla parte client, l'interesse di gruppi di utenti diversi possono essere descritti come viste di un modello di impresa generale.

Nell'implementazione della prospettiva concettuale, nel meta database, la classe centrale è il Modello. Un modello è relazionato con una fonte dati, un client o con una rilevante sezione dell'impresa e rappresenta i concetti che sono disponibili nelle fonti dati, client e impresa rispettivamente. Le classi ClientModel, SourceModel, ed EnterpriseModel sono necessari per distinguere i modelli dei rispettivi livelli. Un modello è composto da Concetti, ognuno dei quali rappresenta un concetto del mondo reale, cioè il mondo del business. Se gli utenti forniscono alcune informazioni riguardanti la relazione tra concetti in un linguaggio formale come la descrizione logica, una persona può ragionare sulle assunzioni fatte per un determinato concetto.

I risultati del processo di ragionamento sono immagazzinati nel modello come attributo "isSubsumedBy" del concetto corrispondente. Essenzialmente, il repository può essere utilizzato come memoria per i risultati del processo di ragionamento. Qualsiasi programma può interrogare il repository per ottenere le informazioni sui concetti; se i risultati sono già stati immagazzinati allora il repository potrà rispondere direttamente dando le informazioni necessarie, altrimenti verrà richiesto al programma di calcolare in risultato.

1.2.2. Prospettiva Logica

La prospettiva logica concepisce un DWH dal punto di vista dei modelli dati coinvolti, cioè il modello dati dello schema logico è dato dai corrispondenti componenti fisici, che implementano lo schema logico. Il punto principale nella prospettiva logica è lo Schema. Così come un modello (descritto nel paragrafo precedente) è composto da concetti, lo schema è composto da Tipi.

Come nella prospettiva concettuale, distinguiamo la prospettiva logica tra ClientSchema, DWSchema, ed SourceSchema, per gli schema del client, del DWH e delle fonti dati.Pe rogni modello dei client o delle fonti dati esiste uno schema. Questa restrizione viene garantita da una costrizione nel modello dell'architettura. Il legame con il modello concettuale avviene tramite la relazione tra concetto e tipo; ogni tipo è espresso come vista di un concetto.

1.2.3. Prospettiva fisica

L'industria dei DWH hanno esplorato in maniera molto più approfondita la prospettiva fisica, tanto che la maggior parte degli aspetti di questa vengono presi dalle analisi di soluzioni commerciali di DWH. I componenti fisici base in un'architettura DWH sono gli agents e i data store. Gli Agents sono programmi che controllano altri componenti oppure che trasportano dati da un luogo fisico ad un altro (per esempio gli ETL). I data store sono database che immagazzinano i dati provenienti da altri componenti.

La classe base nella prospettiva fisica sono i DW_Component. Questi componenti possono essere costituiti, a loro volta, da altri componenti. Ciò viene espresso dall'attributo hasPart. Inoltre, un componente consegna (delivers to) a un altro componente un tipo (Type), che fa parte della prospettiva logica. Un altro legame col modello logico è dato dall'attributo hasSchema del DW_Component. Si noti che un componente può avere uno schema, cioè un set di tipi, ma può consegnare solamente un tipo ad un altro componente.

Ci sono due tipi di Agenti: gli agenti di controllo (ControlAgent), che controllano altri componenti ed agenti (ad esempio può avvisare un altro agente che deve cominciare i processi di update), e gli agenti di trasporto (TransportationAgent), che trasportano dati da un componente all'altro. Una agente può anche notificare errori oppure il termine del proprio processo. Un DataStore immagazzina fisicamente i dati che sono descritti dai modelli e dagli schemi delle prospettive concettuali e logiche rispettivamente. Come nelle altre prospettive, distinguiamo tra ClientDataStore, DW_DataStore e SourceDataStore per i client, DWH e sorgenti rispettivamente.

Di seguito il repository descritto nei paragrafi precedenti.

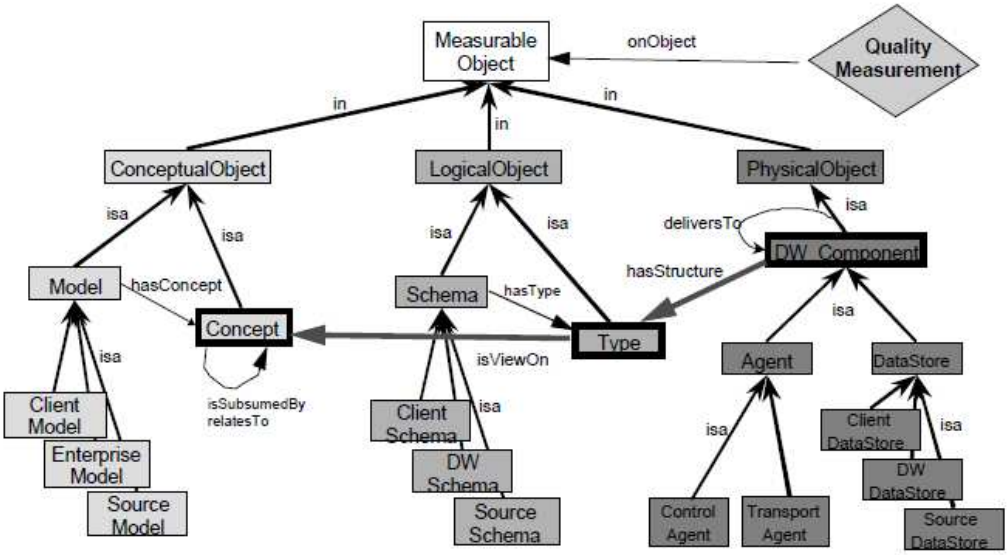


Figura 4 - Struttura del repository del meta modello (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

2. Gestire la qualità di un Data Warehouse

Il problema della qualità dei dati nei Data Warehouse è sicuramente quello più sentito a livello del business di un'azienda; infatti, una poca qualità nei dati può portare a prendere decisioni errate che si possono tradurre in gravi perdite economiche (Wang & Strong, 1996). Quello che serve è, quindi, riuscire ad estendere i modelli di architettura di un DWH per supportare modelli espliciti di qualità dei dati.

In questo capitolo illustreremo come estendere il modello dell'architettura del DWH, a cui siamo giunti nel capitolo precedente, per favorire modelli espliciti di qualità.

Per prima cosa dobbiamo risolvere due questioni fondamentali:

- La qualità di un dato è un fenomeno soggettivo e può cambiare da utente a utente; ciò fa si che si debbano organizzare obiettivi di qualità diversi a seconda dei gruppi di interesse.
- Non esiste un modello assoluto di qualità del dato, pertanto risulta complesso ottenere un misura diretta del risultato.

Il problema generale di introdurre modelli di qualità nei meta-dati è quindi quello di ottimizzare il compromesso tra la realizzazione di una copertura più ampia possibile dei dati e l'utilizzo di conoscenze dettagliate che già si hanno per determinati criteri (Hinrichs, 2000).

Di seguito, nella figura 5, vengono rappresentati quelli che sono i fattori di qualità dei dati più rilevanti in un DWH, secondo quanto indicato dalla maggior parte degli stakeholders.

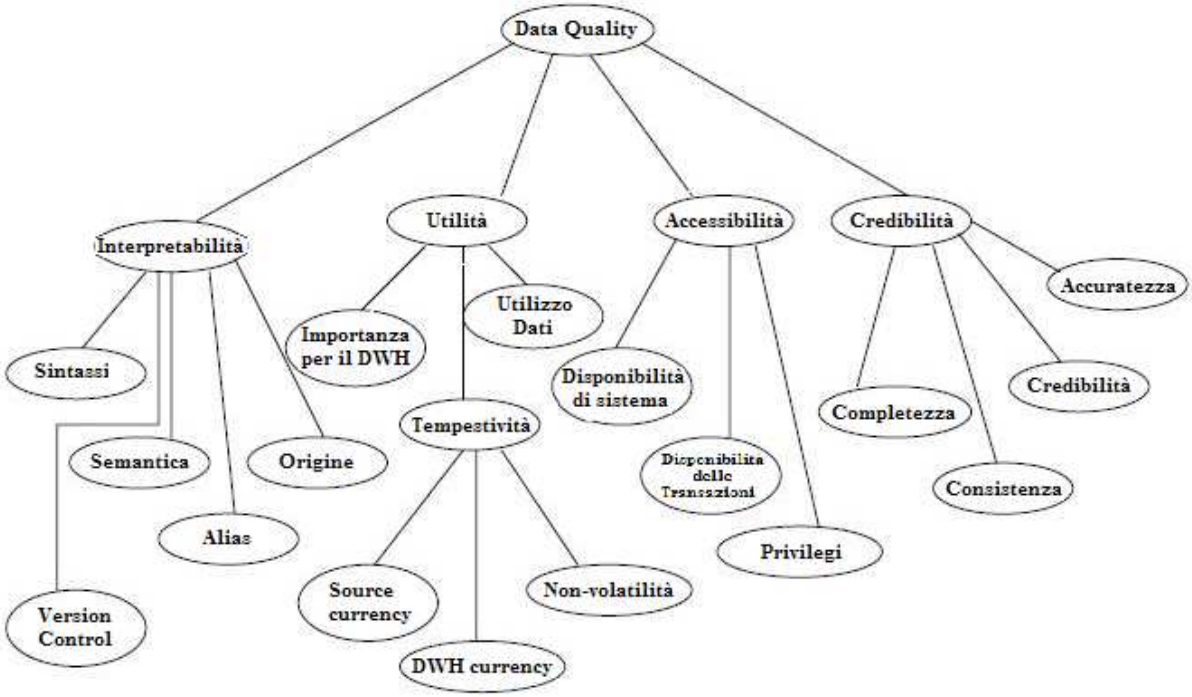


Figura 5 - Fattori di qualità di un DWH (Jarke & Vassiliou, Data Warehouse Quality: A Review of the DWQ Project, 1997)

Conseguentemente, il problema che si pone è la capacità di misurare e creare un modello di qualità del DWH. Poiché il DWH è un sistema composto da diversi sottosistemi e processi che interagiscono tra loro, bisogna avere una corrispondenza tra i componenti del DWH e i modelli di qualità del dato che introdurremo. Inoltre, dobbiamo porci l'obiettivo di sviluppare delle misure per gli indicatori di qualità e una metodologia per progettare un modello di qualità per uno specifico DWH.

Un esame più attento della gerarchia dei fattori di qualità, rivela parecchie relazioni tra i parametri di qualità e gli aspetti progettuali/operativi dei DWH. Nella figura 6 possiamo notare alcune di queste relazioni.

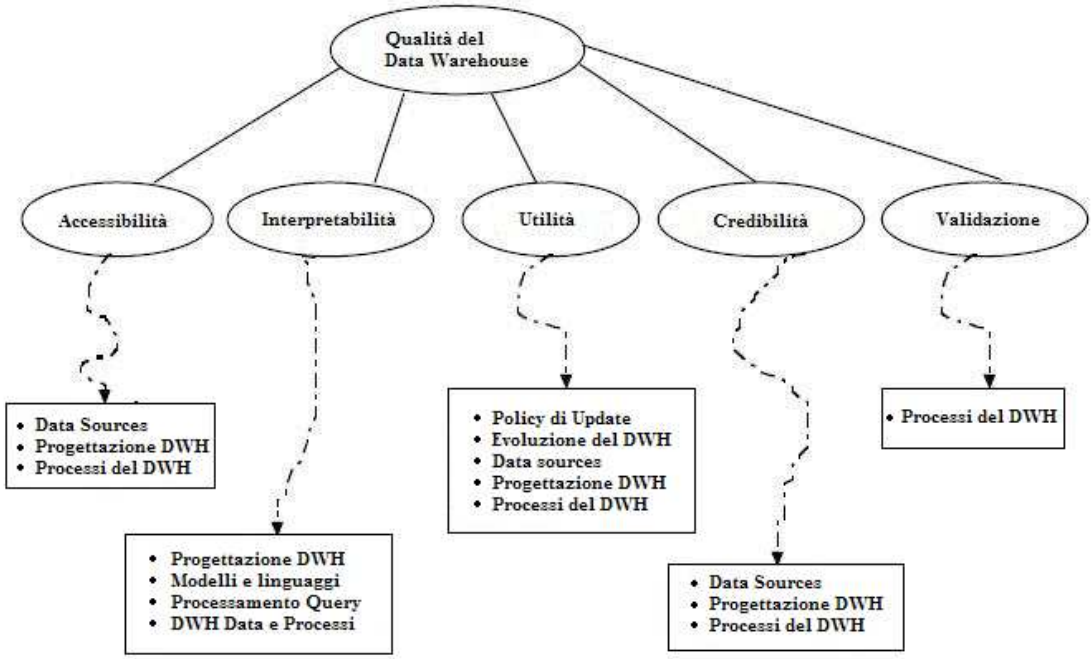


Figura 6 - Fattori di qualità legati ai processi del DWH (DWQ – Foundations of Data Warehouse Quality)

Come abbiamo già indicato precedentemente, la qualità dei dati è un fenomeno soggettivo e viene percepita il maniera diversa a seconda del tipo di utente del DWH; in particolare possiamo identificare:

- Gli utenti di business. Per questo tipo di utenti la qualità è da intendersi come la bontà del dato, il poter avere tempestivamente le informazioni richieste e la facilità con cui poter estrarre queste informazioni. Gli utenti di business, solitamente, utilizzano dei programmi per analisi OLAP.
- Gli amministratori del DWH. Questi utenti necessitano informazioni su eventuali errori del sistema, avere una ottima accessibilità ai meta-dati e devono conoscere le scadenze e le tempistiche necessarie per gli utenti di business per trovare le cause di cambiamenti e problemi nel DWH.
- Gli sviluppatori del DWH. Questi utenti hanno bisogno di poter misurare la qualità dei modelli dell'ambiente del Data Warehouse e la bontà dei dati.

In base a ciò, possiamo tranquillamente affermare che le differenti tipologie di utenti che utilizzano il DWH portano ad un insieme di "dimensioni" di qualità, che un modello deve essere in grado di indirizzare in modo significativo.

Qui di seguito vedremo in maggior dettaglio le dimensioni di qualità, già sopra indicate, divise a seconda delle tre tipologie di utenti che abbiamo preso sopra in considerazione: Gli amministratori del DWH, i programmatori e gli utenti di Business.

Qualità di progetto e amministrazione (amministratori e programmatori)

Questa dimensione di qualità può essere analizzata suddividendola in parti di maggior dettaglio come indicato nella figura 7.

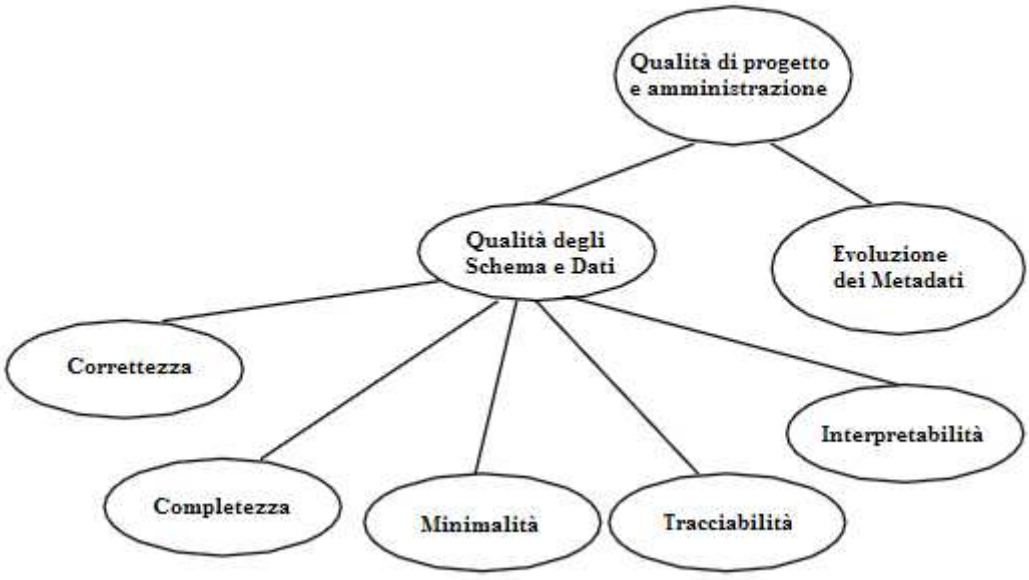


Figura 7 - Dimensione "Qualità di progetto e amministrazione" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Qualità degli schemi e dei dati si riferisce di uno schema o di un modello di rappresentare adeguatamente ed efficientemente le informazioni; lo stesso criterio si può anche applicare livello di

istanza dei dati. La dimensione *Correttezza* si occupa della corretta comprensione delle entità del mondo reale, degli schemi e dei bisogni degli utenti. La dimensione *Completezza* si occupa di mantenere tutte le informazioni e le conoscenze critiche all'interno degli schemi del data warehouse. La dimensione *Minimalità* descrive il grado di ridondanza indesiderata da evitare durante il processo di integrazione delle informazioni delle fonti dati. La dimensione *Tracciabilità* si occupa di verificare che tutti i tipi di requisiti degli utenti, sviluppatori, amministratori e manager siano tracciati nello schema del DWH. La dimensione *Interpretabilità* assicura che tutti i componenti del DWH siano ben descritti, in maniera tale da essere amministrati con facilità. La dimensione evoluzione dei metadati si occupa di come gli schemi si evolvono durante l'operatività del DWH. La tabella 1 mette in relazione le dimensioni di qualità con gli oggetti del data warehouse e mostra come la qualità di questi oggetti possa essere misurata.

Qualità di progetto e amministrazione	**Prospettiva concettuale**		**Prospettiva logica**	
	Modello	**Concetto**	**Schema**	**Tipo**
Correttezza	Numero di conflitti con gli altri modelli/mondo reale	Correttezza della descrizione delle entità del mondo reale	Correttezza di mappatura dei modelli concettuali con gli schemi logici	Correttezza di mappatura dei concetti in tipi
Completezza	Livello di copertura, numero di regole di business rappresentate	Numero di attributi mancanti; Sono complete le asserzioni relative al concetto?	Numero di entità mancanti rispetto al modello concettuale	Numero di attributi mancanti rispetto al modello concettuale
Minimalità	Numero di entità o di relazioni ridondanti nel modello	Equivalenza nella descrizione delle stesse cose in concetti diversi per lo stesso modello	Numero di relazioni ridondanti	Numero di attributi ridondanti
Tracciabilità	I requisiti e i cambiamenti degli sviluppatori sono salvati?	I requisiti e i cambiamenti degli sviluppatori sono salvati?	I requisiti e i cambiamenti degli sviluppatori sono salvati?	I requisiti e i cambiamenti degli sviluppatori sono salvati?
Interpretabilità	Qualità della documentazione	Qualità della documentazione	Qualità della documentazione	Qualità della documentazione
Evoluzione dei Metadati	È documentata l'evoluzione del modello?	È documentata l'evoluzione del concetto?	È documentata l'evoluzione dello schema?	È documentata l'evoluzione del tipo?

Tabella 1 - Esempio di misure per la dimensione "Qualità di progetto e amministrazione" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Qualità di Implementazione del software (programmatori)

L'implementazione e la valutazione del software non sono compiti con specifici per un data warehouse; pertanto, non si vuole proporre un nuovo modello per queste funzioni, ma adotteremo quelle definite nello standard ISO 9126.

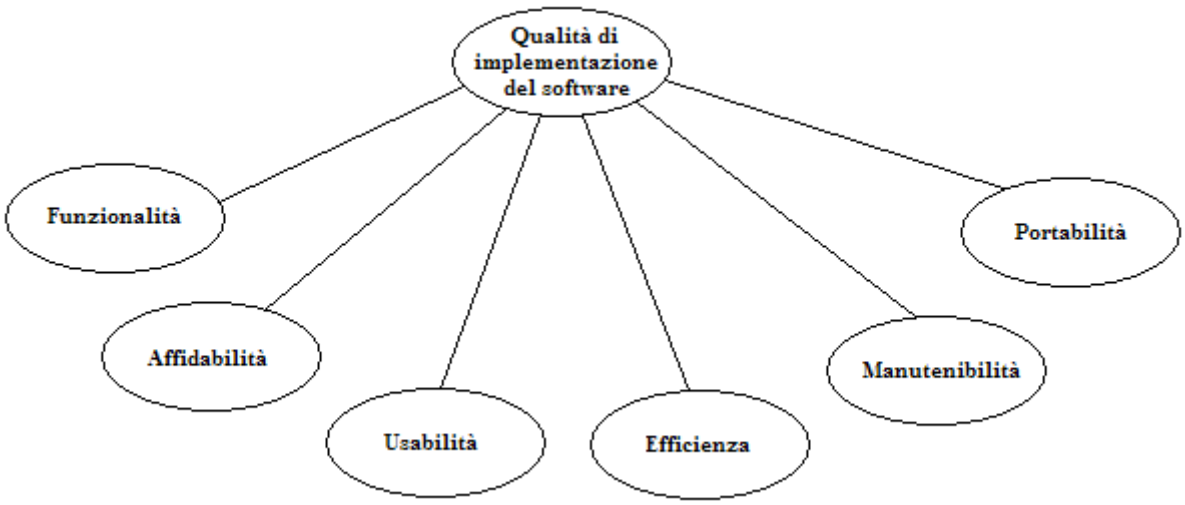


Figura 8- Dimensione "Qualità di implementazione del software" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Le dimensioni sono *Funzionalità* (idoneità, accuratezza, interoperabilità, compliance, sicurezza), *Affidabilità* (maturità, tolleranza ai malfunzionamenti, recuperabilità), *Usabilità* (comprensibilità, facilità di apprendimento, operabilità), *Efficienza* (in termini di utilizzo di tempo e di risorse), *Manutenibilità* (capacità di essere analizzato, mutevolezza, stabilità, testabilità), *Portabilità* (adattabilità, facilità di installazione, conformità, sostituibilità).

Questa dimensione di qualità è applicabile solo alla prospettiva fisica dell'architettura dove il software (agent e data source) è rappresentato. La tabella 2 indica alcuni esempi di come possa essere misurata la qualità per ogni specifico componente.

Qualità di implementazione del software	Prospettiva Fisica
	Componenti del DWH
Funzionalità	Numero di funzioni non appropriate per determinati compiti, numero di moduli incapaci di interagire con specifici sistemi
Affidabilità	Frequenza di errore, tolleranza ai malfunzionamenti
Usabilità	Approvazione degli utenti
Efficienza	Performance, tempi di risposta, tempi di elaborazione
Manutenibilità	Ore uomo necessarie per la manutenzione e il test del software
Portabilità	Numero di casi in cui il software fallisce nella migrazione ad un nuovo ambiente; ore uomo necessarie per installare il software nel nuovo ambiente

Tabella 2 - Esempi di misure di qualità per la dimensione "implementazione del software" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Qualità dell'utilizzo dei dati (amministratori e utenti di business)

Poiché i database e, nel nostro caso, i data warehouse sono costruiti in modo tale da poter essere interrogati, il processo base per il DWH è l'utilizzo e l'interrogazione dei propri dati. Nella figura 9 viene raffigurata la gerarchia riguardo la dimensione di qualità dell'utilizzo dei dati. Analizziamo ora nel dettaglio ogni singola dimensione.

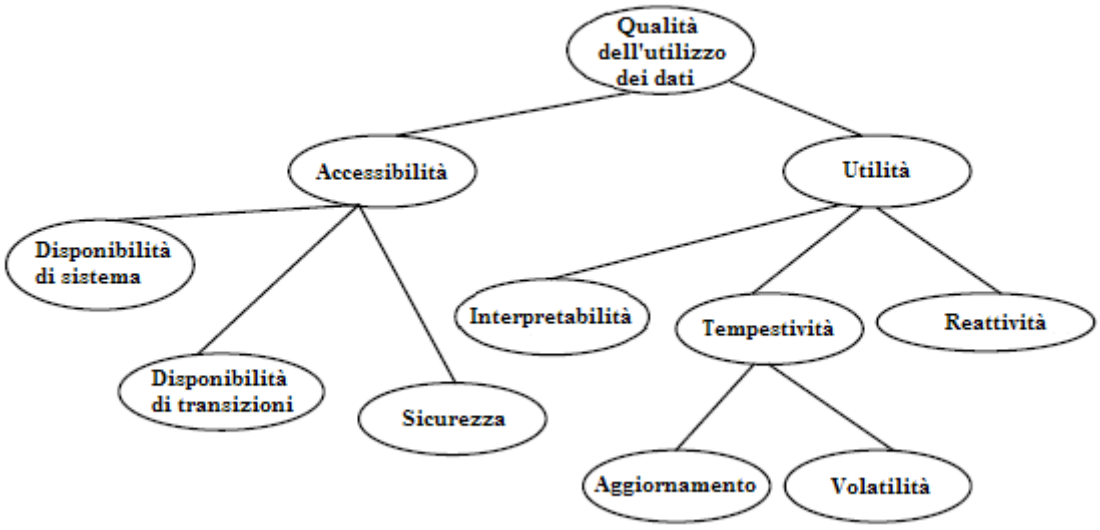


Figura 9 - Dimensione "Qualità dell'utilizzo dei dati" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

La dimensione accessibilità è relativa alla possibilità di raggiungere i dati per poterli interrogare. La dimensione sicurezza descrive la politica di autorizzazione e i privilegi che ogni utente ha per l'interrogazione dei dati. La disponibilità di sistema descrive la percentuale di tempo in cui le fonti dati o il DWH sono disponibili (cioè il sistema è su e non ci sono backup in corso). La dimensione disponibilità delle transazioni, come già descritto, descrive la percentuale di tempo in cui le informazioni presenti nel DWH o nelle fonti dati sono disponibili grazie all'assenza di processi di update che bloccano i dati in scrittura. La dimensione *Utilità* descrive le caratteristiche temporali (*tempestività*) dei dati così come la *reattività* del sistema. La *reattività* si occupa dell'interazine dei processi con l'utente (ad esempio un programma di query che riporta a se stesso quanto tempo è necessario per risolvere la query). La dimensione *Aggiornamento* descrive quando le informazioni sono state inserite nelle fonti dati o/e nel DWH. La dimensione *Volatilità* descrive il periodo di tempo per il quale l'informazione è valide nel mondo reale. La dimensione *Interpretabilità*, come già menzionato, descrive il grado con cui il DWH è modellato efficientemente nel repository delle informazioni. Migliore è la spiegazione, più facilmente potranno essere poste le interrogazioni. Nella tabella 3 vengono mostrati alcuni esempi di come possa essere misurata la qualità dell'utilizzo dei dati.

Qualità dell'utilizzo dei dati	**Prospettiva logica**		**Prospettiva fisica**	
	Schema	**Tipo**	**Agent**	**Data Store**
Accessibilità	Lo schema delle definizioni è accessibile agli utenti?	Il tipo è visibile e accessibile agli utenti?	La rete è sufficiente per i dati consegnati?	Il data store è accessibile?
Disponibilità	Frequenza di update	Frequenza di update	Tempo di risposta	Tempo di update del data store, tempo di risposta
Sicurezza	Livello di sicurezza (diritti di accesso)	Livello di sicurezza (diritti di accesso)	Ci sono restrizioni fisiche per l'accesso?	Il data store è capace di bloccare accessi non autorizzati?
Utilità	Lo schema viene utilizzato da qualche utente?	Il tipo viene utilizzato da qualche utente?	I dati trasportati dall'agent vengono veramente utilizzati nello store destinazione?	I dati di questo store vengono interrogati da qualcuno?
Interpretabilità	Lo schema è comprensibile?	Il tipo è comprensibile?	Il dato consegnato è comprensibile?	Il dato immagazzinato è comprensibile?

Tabella 3 - Esempi di come misurare la qualità delll'utilizzo dei dati (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Qualità dei dati (utente di business)

La qualità dei dati che sono immagazzinati in un DWH, ovviamente, non è un processo a sé stante; infatti è influenzata da tutti i processi che fanno parte dell'ambiente del data warehouse.

Definiamo la qualità dei dati come un piccolo sottoinsieme delle dimensioni proposte negli altri modelli. Le dimensioni base che andiamo a descrivere sono mostrate nella figura 10.

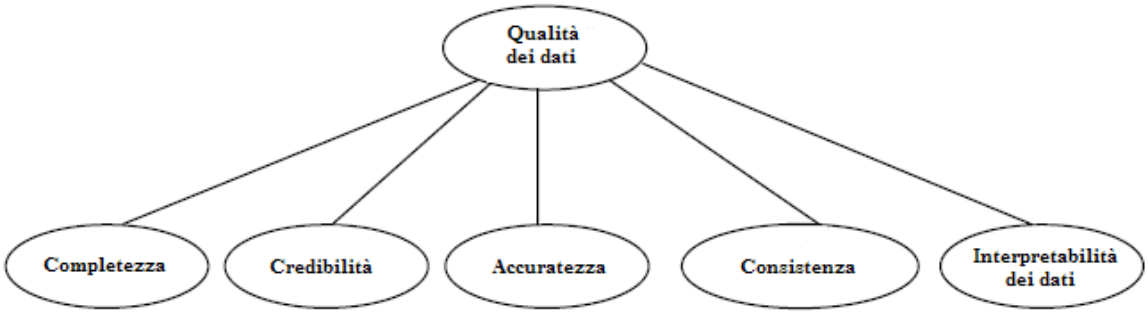


Figura 10 - Dimensione "Qualità dei dati" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

La dimensione *qualità dei dati* non copre un processo del DWH: si riferisce direttamente alle proprietà dei dati immagazzinati (cioè né dello schema né del modello). Per questo motivo viene collegata alla prospettiva fisica dell'architettura rappresentando i data store e gli agent di tutti i livelli. Vediamo ora in dettaglio queste dimensioni.

La dimensione *completezza* descrive la percenuale di informazioni del mondo reale che sono entrate nelle fonti dati o/e nel data warehouse. Per esempio, la completezza può valutare il grado con cui una stringa che descrive un indirizzo abbia effettivamente il formato e la lunghezza corretta per contenere l'informazione.

La dimensione *credibilità* descrive l'affidabilità (non in termini di disponibilità, bensì di correttezza delle informazioni) della fonte dati che fornisce le informazioni. La dimensione *accuratezza* descrive, appunto, l'accuratezza del processo di inserimento dati che avviene alla fonte. La dimensione *consistenza* descrive la coerenza logica delle informazioni.

La dimensione *interpretabilità dei dati* si occupa della descrizione di questi ultimi (cioè il layout dei dati per i sistemi legacy e dati esterni, descrizione delle tabelle per i database relazionali, chiavi primarie ed esterne, alias, settaggi di default, domini, spiegazione dei valori dei codici, ecc…).

Alcune metriche per la qualità dei dati sono date in tabella 4

Qualità dei dati	**Prospettiva Fisica**	
	Agent	**Data Store**
Completezza	Numero di tuple consegnate rispetto al numero atteso	Numero di valori a null dove non sono attesi
Credibilità	Credibilità del processo che consegna i valori	Numero di tuple con i valori di default
Accuratezza	Numero di tuple consegnate esatte	Livello di precisione: numero di tuple consegnate esatte
Consistenza	I dati consegnati sono consistenti rispetto ad altri dati?	Numero di tuple che violano costrizioni, numero di differenze di codifica
Interpretabilità dei dati	Numero di tuple con dati interpretabili, documentazione per i valori chiave, il formato è comprensibile?	Numero di tuple con dati interpretabili, documentazione per i valori chiave, il formato è comprensibile?

Tabella 4 - esempio di misure per la dimensione "Qualità dei dati" (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

2.1. Problema di valutazione della qualità in multi-criteri eterogenei

Passiamo ora al trattamento formale e alla gestione basata su repository degli obiettivi di qualità del DWH descritti nella sezione precedente.

Una prima formalizzazione potrebbe essere basato su un'analisi qualitativa dei rapporti tra gli stessi fattori di qualità ad esempio rapporti obiettivi-secondi fini positivi o negativi o rapporti obiettivi-significati. Gli utenti di business possono quindi inserire la loro valutazione soggettiva di obiettivi individuali, nonché dare dei pesi possibili agli obiettivi ed essere supportati per individuare buoni compromessi. Le valutazioni inserite e calcolate sono utilizzate come misura di qualità nel modello di architettura di figura 3, consentendo quindi un'integrazione molto semplice tra l'architettura e il modello di qualità.

Tale approccio è ampiamente usato in ingegneria industriale sotto l'etichetta di *distribuzione della funzione di qualità*, utilizzando un particolare tipo di rappresentazione matriciale chiamato la "House of Quality".

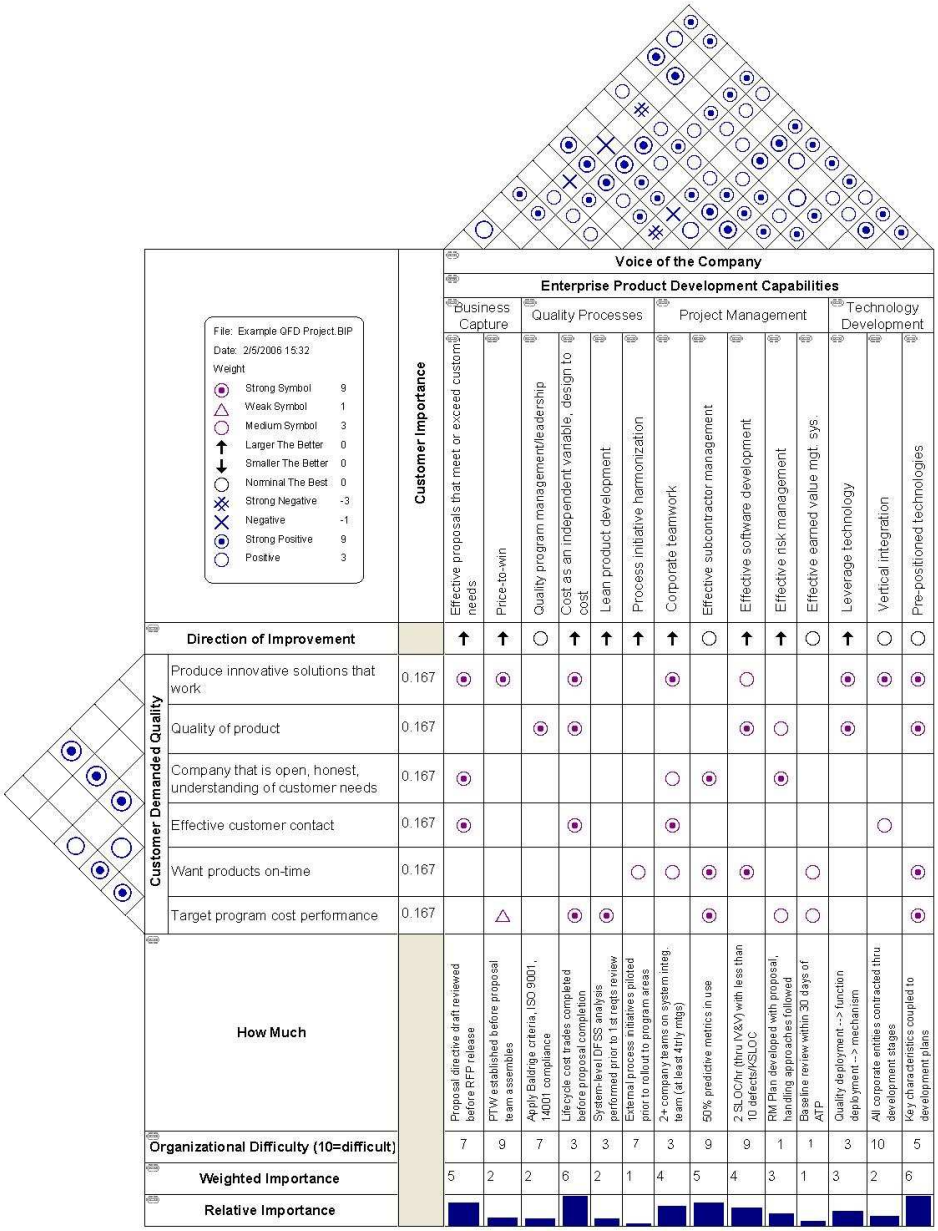


Figura 11 - Esempio di matrice "House of quality"

Tuttavia, mentre questo approccio semplice ha certamente un ruolo utile nel prendere decisioni a fronte di una moltitudine di criteri da valutare, utilizzando da sola vorrebbe dire buttare via la ricchezza del lavoro creato dalla ricerca di misura, di previsione o di ottimizzazione di singoli fattori di qualità del DWH. Per dare un'idea della ricchezza delle tecniche da prendere in considerazione, si usa un unico fattore di qualità – la reattività nel senso di rapidità di esecuzione

delle query - per il quale studiamo tre approcci diversi, uno per le prospettiva concettuale, uno per quella logica e uno per quella fisica.

Cominciamo dalla prospettiva logica. In questo caso, la misura della qualità associata alla reattività viene considerata come una media pesata dei "costi" delle query e degli aggiornamenti per un dato insieme di query e di fonti dati. Una tecnica di ottimizzazione combinatoria che viene proposta è quella di creare un insieme di viste materializzate per minimizzare il costo totale. Questa soluzione può essere considerata come un caso molto semplice di approccio di distribuzione della funzione di qualità, ma ha in più il vantaggio di una progettazione automatizzata della soluzione.

Se includiamo la prospettiva fisica, la definizione di "costo" delle query e degli aggiornamenti diventa un problema di per sé: Cosa intendiamo per costo – tempo di risposta, rendimento oppure una combinazione dei due (per esempio minimizzare il tempo di risposta delle interrogazioni e massimizzare il rendimento degli aggiornamenti)? Qual è la causa di questi costi – è l'accesso al database o il traffico della rete il collo di bottiglia? Un modello globale di gestione delle code permette di capire a fondo i dettagli di tali metriche dalle quali lo sviluppatore potrà scegliere quelle migliori per gli obiettivi di qualità del processo che sta progettando. Inoltre, si mettono in gioco opzioni completamente nuove nel processo di sviluppo: invece di materializzare più viste per migliorare i tempi di risposta delle query (al costo di aumentare il tempo necessario per l'aggiornamento dei sistemi OLTP), il progettista può decidere di compare un PC o un database più veloce, oppure aumentare la velocità di connessione della rete.

Eppure altre opzioni entrano in gioco, quando è disponibile una ricca logica per il trattamento delle prospettive concettuali. Per esempio, la descrizione logica sviluppata per l'integrazione delle fonti dati permette di affermare che le informazioni riguardo tutte le istanze di un concetto del modello di impresa, sono mantenute in una particolare fonte dati. In altre parole, la fonte dati è completa rispetto al dominio. Ciò permette agli sviluppatori del DWH di droppare tutte le viste materializzate nelle altre fonti dati, riducendo quindi riducendo sistematicamente lo sforzo necessario per gli aggiornamenti senza perdere completezza delle risposte.

2.2. Valutazione gerarchica di qualità: un approccio GQM adattato

È chiaro che non può esistere un oggettivo quadro formale che copra tutti questi aspetti con un linguaggio uniforme. Quando progettiamo le estensioni al meta database per la gestione della qualità; dobbiamo, quindi, cercare un'altra soluzione che mantenga ancora il quadro complessivo offerto dalle tecniche superficiali di gestione della qualità come la *distribuzione della funzione di qualità* ma, allo stesso tempo, essere aperti all'inserimento di valutazioni e tecniche di progettazione specializzate.

La soluzione proposta per questo problema si basa sul diffuso approccio GQM (Goal-Question-Metric) utilizzato nella gestione di qualità del software. L'idea del GQM è che *obiettivi* (Goal) di qualità non possono essere valutati direttamente. Invece, il loro significato è circoscritto da *domande* (Question) che devono ottenere risposte mentre si valuta la qualità. Le domande relative alla qualità, di nuovo, solitamente non possono avere delle risposte dirette, ma devono appoggiarsi a delle *metriche* (Metrics) applicate o al prodotto o al processo in questione; tecniche come grafici statistici di controllo del processo vengono poi applicate per ricavare la risposta ad una domanda partendo dalla misurazione.

Nell'esempio riportato al paragrafo precedente, l'*obiettivo* della reattività può essere raffinato in *domande* riguardo al compromesso tra le prestazioni delle interrogazioni a db e i processi di aggiornamento (prospettiva logica), riguardo alla presenza di colli di bottiglia al livello fisico, e a riguardo della completezza o anche della ridondanza dei data source utilizzati (prospettiva concettuale). Queste domande possono essere risposte utilizzando le *metriche* e gli *algoritmi* sopra menzionati.

La soluzione di repository che utilizzeremo usa un approccio simile per colmare il divario tra le gerarchie di obiettivi di qualità da una parte, e tecniche di ragionamento e metriche molto dettagliate dall'altra. Il collegamento è definito attraverso l'idea di interrogazioni di qualità come viste materializzate sul data warehouse; le viste sono definite attraverso generiche queries sulla misurazione della qualità.

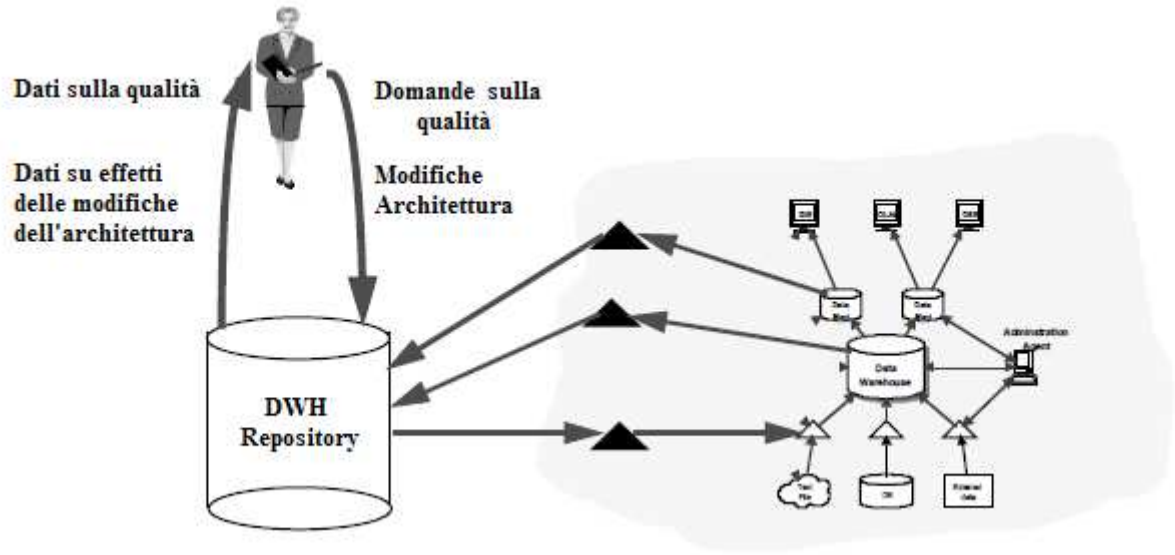


Figura 12 - Gestione della qualità attraverso il repository del DWH (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

La figura 12 motiva questo approccio ponendo attenzione al repository. Gli utenti di business valutano la qualità del DWH interrogando il repository tramite query riguardanti, appunto, la qualità. Il repository risponde alle interrogazioni accedendo ai dati relativi alla qualità ottenuti dagli agenti che si occupano della misura della stessa (i triangolini neri mostrati in figura). Gli agenti comunicano con i componenti del DWH reale per estrarre i valori delle misure.

Gli utenti di business possono ridefinire i propri obiettivi di qualità tutte le volte che vogliono. Ciò porterà ad un aggiornamento del modello di qualità presente nel repository e, possibilmente, alla configurazione di nuovi agenti responsabili di fornire i nuovi dati di qualità. Allo stesso modo, un utente di business che ha determinate autorizzazioni può ridefinire l'architettura del DWH tramite il repository. Un tale aggiornamento evolutivo, per esempio la definizione di un nuovo data source, porta alla riconfigurazione del DWH reale. In ultima analisi, le misure della qualità rifletteranno gli effetti del cambio daranno evidenza del fatto che l'evoluzione abbia portato o meno a miglioramenti di qualche obiettivo di qualità.

L'utilizzo del repository per la gestione della qualità di un data warehouse ha vantaggi significativi:

- I sistemi DWH contengono già repository per gestire i metadati riguardanti il DWH; estendere questi componenti per la gestione della qualità è un passo naturale.

- Meta data esistenti riguardanti il DWH, per esempio la staging area, possono essere utilizzati direttamente per formulare obiettivi di qualità e piani di misurazione
- Il modello di qualità può essere ritenuto coerente con il modello dell'architettura, cioè il repository può prevenire la formulazione di obiettivi di qualità, da parte degli utenti di business, che non possono essere validati con l'architettura esistente.
- Gli utenti di business accedono al repository come un data source per fonrine report di qualità a quegli utenti che formulano gli obiettivi; infatti, la produzione di tali report è lo stesso tipo di attività che viene utilizzata per fornire dati aggregati ai programmi client di un DWH.

L'ultimo punto non è solo un commento tecnico. I dati di qualità, vale a dire i valori delle misure di qualità, sono ricavati da componenti del DWH. I valori sono viste materializzate proprie di questi componenti. Questi valori hanno loro stessi valori come tempestività e accuratezza. È differente se i valori di una misura di qualità vengono aggiornata ogni ora o una volta al mese. Anche se non entriamo nel dettaglio di questo "secondo livello" di qualità, notiamo che gli stessi metodi che vengono utilizzati per mantenere la qualità del DWH possono essere usati anche per mantenere la qualità del repository del DWH (che ospita il modello di qualità).

2.3. Il meta modello della qualità

I dati riguardanti la qualità dono dati derivati e sono mantenuti dal sistema DWH. Questa strategia di implementazione fornisce maggior supporto tecnico rispetto a implementazioni di tipo GQM per sistemi software in generale. Il linguaggio espressivo di interrogazione offerto dal sistema del repository di concetti base (ConceptBase), rende la gran parte dei compiti di gestione di qualità una questione di formulazione di query. Qui di seguito verrà descritto come una versione di GQM può essere modellata tramite le meta classi di Telos nel ConceptBase e come può essere utilizzato per la formulazione di obiettivi di qualità e analisi di qualità.

Telos fornisce una rappresentazione logica dell'appartenenza alle classi (*x è nella classe*), specializzazione tra classi (*c è una d*), e attributi (*x è etichetta di y*). Questa rappresentazione logica può essere mappata in un layout grafico come mostrato per il modello di qualità qui sotto, così

come può esserlo in una sintassi strutturata che a volte utilizziamo per la formulazione di queries. Dato che tutto (oggetti, classi, meta classi ed attributi) sono trattati uniformemente nella rappresentazione logica, il linguaggio di Telos viene utilizzato, estendendo l'approccio mostrato nella figura 5, per formulare

- Un meta modello da un insieme di meta classi (per definire l'architettura e il modello di qualità)
- Un insieme di classi (utilizzando l'architettura e meta modelli di qualità per esprimere gli obiettivi di qualità, interrogazioni, e tipi di misura sui componenti del DWH)
- Istanze di classi (per rappresentare i risultati delle misure come istanze di classi)

I sistemi di data warehouse sono unici, nel senso che si basano meta database (o repository) run-time che immagazzina informazioni riguardo ai dati e ai processi del sistema. Questa cosa apre all'opportunità di implementare l'approccio GQM in modo tale che si riferisca direttamente ai concetti del meta database del DWH.

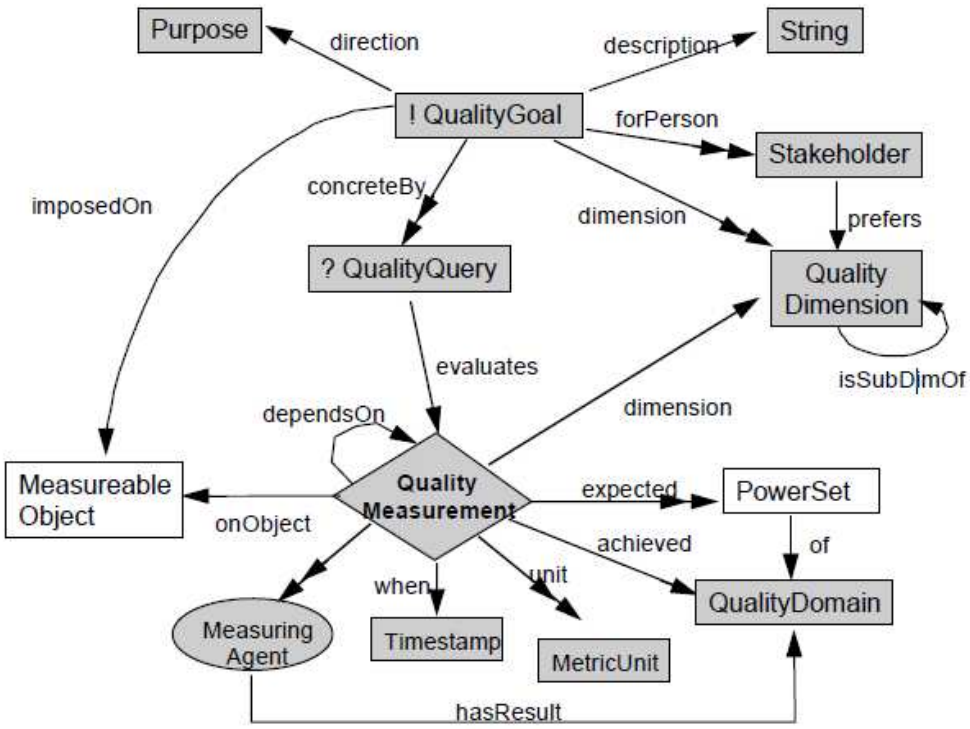


Figura 13 - Meta modello per la qualità nel DWH (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

La figura 13 mostra le meta classi di Telos per la gestione della qualità nel DWH. Gli obiettivi della qualità, per esempio “migliorare la tempestività del data set vendite per mese”, vengono assegnati agli utenti di business. L’attributo scopo (*Purpose*) per gli obiettivi di qualità viene utilizzato per specificare il percorso previsto per il miglioramento della qualità (ad esempio, per aumentare il livello di qualità o per raggiungere un determinato livello di qualità in un intervallo di tempo definito). L’obiettivo di qualità è dato da oggetti misurabili del DWH (*imposedOn*) come classificato dal modello di architettura della figura 4. Gli obiettivi di qualità (*Quality Goal*) sono mappati su una serie di query di qualità (*Quality Query*) che vengono utilizzate per decidere se un obiettivo è realizzato o meno. Nella nostra versione di GQM , queste interrogazioni vengono effettuate sul repository del DWH. Un obiettivo è collegato ad una o più dimensioni di qualità (*Quality Dimension*) a seconda delle preferenze degli utenti di business che creano l’obiettivo.

Il prossimo concetto chiave è la *Quality Query*. Mentre nell’approccio GQM originale è considerata solo un testo, noi la traduciamo come una query eseguibile sul repository del DWH utilizzando il linguaggio espressivo deduttivo del ConceptBase. La risposta a un’interrogazione della qualità è vista come prova per il conseguimento di un obiettivo di qualità. Il più semplice tipo di query di qualità non farà altro che valutare se la misura di qualità (*Quality Measurement*) presa in considerazione per un oggetto del DWH è compresa nell’intervallo di valori atteso. Una misura di qualità utilizza una unità metrica, per esempio il numero medi di valori nulli per una tupla di relazioni.

2.4. Supporto per l’implementazione del meta modello di qualità

I livelli di astrazione dei concetti nel modello di qualità richiedono un esame più attento. Nelle metriche dei software standard, la *misura di qualità* è una funzione che mappa un’entità del mondo reale con un valore di un dominio, solitamente a un numero. Nel nostro caso, manterremo una rappresentazione astratta di tutte le entità del mondo reale che riteniamo “interessanti” nel repository del DWH. Pertanto, le misure della qualità possono essere registrati come relazioni esplicite tra le rappresentazioni astratte, cioè oggetti misurabili, e i valori della qualità. Per natura, questo tipo di misura della qualità si riferisce a oggetti di differenti livelli di astrazione.

Per esempio, un valore di qualità di 0.8 può essere misurato come percentuale di valori a null della relazione tra un oggetto *Impiegato* ed alcuni data source. *Impiegato* è una relazione (il tipo delle istanze del data source *Impiegato*) mentre 0.8 è solo un numero. Per questo motivo abbiamo bisogno di una rete tipo quella di Telos che è in grado di mettere in relazione oggetti di diverso livello di astrazione.

Una seconda osservazione deve essere fatta per l'uso del modello di qualità da parte di istanze. Istanze tipiche degli oggetti misurabili (*Measurable Objects*) sono oggetti come *Relazione* (prospettiva logica) o tipologie di entità (prospettiva concettuale). Questi oggetti sono indipendenti dal dominio di applicazione del DWH; vengono usati per descrivere l'architettura di un data warehouse ma non sono componenti concreti di questa architettura. Un'architettura concreta è composta da oggetti come data sources per la relazione *Impiegato,* ETL concreti ecc... Pertanto, quando si istanzia un modello di qualità descriviamo i tipi di obiettivo di qualità, i tipi di interrogazioni e i tipi di misura. Per esempio possiamo descrive un obiettivo di completezza per fonti dati relazionali (istanze del concetto di *Relazione* descritto in figura 4) che è misurato dal conteggio della percentuale di valori nulli nella relazione. Questi tipi possono essere riutilizzati per qualsiasi architettura concreta di un DWH.

I fattori di qualità elencati nelle tabelle dalla 1 alla 4 sono questo tipo di misura e hanno bisogno di essere istanziati da misure concrete. Questo modo di creare istanze in due passi è essenziale nel nostro approccio in quanto permette pre-caricare il repository con obiettivi di qualità, tipi di interrogazioni e misure indipendenti dal dominio dell'applicazione. In altre parole, il repository ha conoscenza dei metodi di gestione della qualità.

Gli obiettivi di qualità, le cui dimensioni sono organizzate in gerarchie come mostrato nelle figure 9,10 e 11, sono resi operazionali come tipi di interrogazioni definite sulle misure di qualità. Queste interrogazioni supporteranno la valutazione di uno specifico obiettivo di qualità quando parametrizzato con una data parte di meta database del DWH. Questo tipo di query, solitamente, confrontano l'obiettivo delle analisi ad un certo intervallo atteso al fine di valutare il livello di qualità raggiunto.

Di conseguenza, la misura della qualità deve contenere entrambi i valori, quello atteso e quello attuale. Entrambi possono essere inseriti nel meta database manualmente, o calcolati induttivamente da una metrica data attraverso un meccanismo di ragionamento specifico. Per esempio, per un dato

progetto fisico ed alcune misure base dei componenti e velocità di rete, un modello di accodamento calcola le misure di qualità di tempo di risposta e il throughput, e può indicare se la rete o l'accesso al db è il collo di bottiglia del sistema nei settaggi attuali. Questi risultati possono essere poi combinati con le misure di qualità dei livelli concettuale e logico per ottimizzare l'obiettivo di qualità sottolineato.

Generalmente parlando, le interrogazioni utilizzate per gli obiettivi di qualità accedono a informazioni registrate dalle misure di qualità. Una misura di qualità contiene le seguenti informazioni riguardo a componenti del DWH:

1. Un intervallo di valori attesi
2. Le misure di qualità raggiunte
3. Le metriche utilizzate per calcolare una misura
4. Dipendenze causali ad altre misure di qualità

Le dipendenze tra misure di qualità possono essere utilizzate per tracciare problemi di qualità, cioè misure che sono al di fuori dell'intervallo atteso, fino alla loro causa.

2.5. Comprendere, controllare e migliorare la qualità con il repository

Riassumendo quanto detto fino ad ora, la figura 14 dà un'idea di come l'architettura tradizionale della figura 1 viene estesa dall'approccio, appena discusso, di gestione centrata su repository di met dati. Il modello di qualità forma le basi per l'implementazione del ConceptBase. I dati di qualità (cioè i valori delle misure) vengono inseriti nel sistema ConceptBase da agenti di misurazione esterni che sono strumenti specializzati in analisi e ottimizzazioni. Nel progetto DWQ vengono sviluppato 4 di questi strumenti; accanto agli strumenti di ragionamento delle assunzioni, già menzionato in precedenza, abbiamo uno strumento per la verifica dell'aggiornamento dei dati, uno per ragionare sugli aggregati multidimensionali e uno per ottimizzare le query lato client. Il ConceptBase può innescare questi agenti in base alle tempistiche associate ad ognuno nel repository.

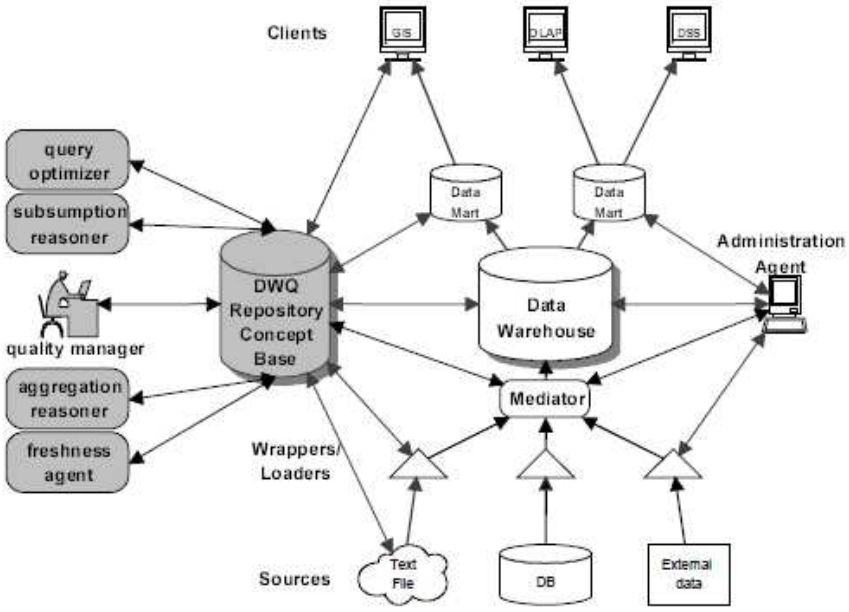


Figura 14 - Modello esteso di architettura del DWH con il modello di qualità (Jarke, Jeusfeld, Quix, & Vassiliadis, 1999)

Il risultato delle analisi dei dati di qualità può essere mostrato graficamente. La rappresentazione grafica aiuta a comprendere immediatamente se ci sono punti in cui bisogna inserire dei controlli ad hoc (gli ovali neri). In questo modo, dato che ogni utente di business ha i propri obiettivi di qualità, sarà possibile visualizzare se i propri obiettivi sono rispettati o meno.

L'ultimo e più avanzato aspetto della gestione della qualità è il *miglioramento*. Il modello descritto fin'ora non contiene un metodo costruttivo su come migliorare la qualità di un DWH. Il primo passo è quello di incorporare il modello matematico descritto sopra, dopodiché sarà compito del progettista apportare cambiamenti incrementali al DWH tali per cui si possano misurare gli effetti sulla qualità.

3. Caso pratico: il DWH di una multinazionale farmaceutica

Nei capitoli precedenti abbiamo descritto quanto sia importante godere di un DWH con dati di qualità; in questo capitolo vedremo come implementare gli schemi teorici fin qui discussi, utilizzando, come esempio, il data warehouse di una multinazionale farmaceutica. Descriveremo l'architettura del DWH in considerazione, definiremo alcune entità del mondo reale che sono "core" per il modello di business e studieremo come migliorare la gestione della qualità di queste entità.

3.1. Architettura del Data Warehouse

L'architettura del DWH che utilizzeremo per il nostro studio è indicata in figura 15.

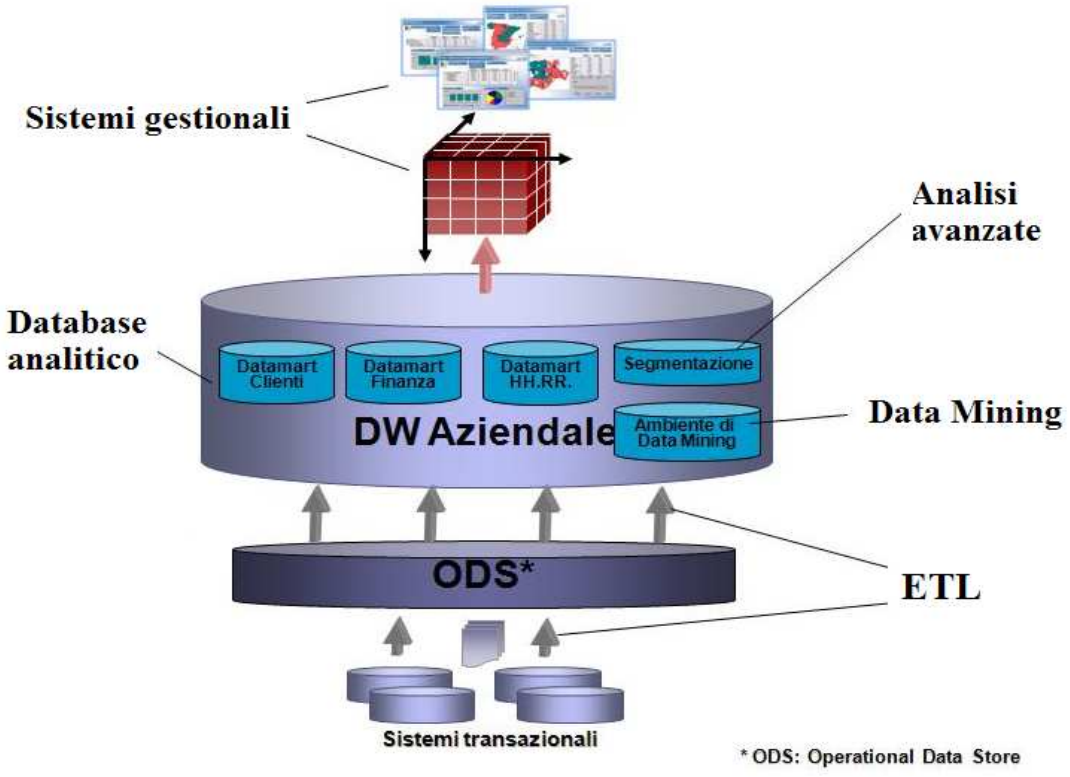


Figura 15 - Struttura del DWH dell'azienda caso di studio

Come possiamo notare, le fonti dati che alimentano le tabelle del sistema DWH sono:

- Il sistema ERP
- Il sistema CRM
- File .xls, .txt, .csv

Le informazioni vengono lette e portate nel primo database (chiamato Staging Area) tramite gli ETL, che hanno il compito di riaggregare i dati in maniera tale da trasportarli in modo corretto per essere immagazzinati dal secondo database, cioè il DWH vero e proprio. Da qui i dati vengono letti dalla suite di Business Intelligence, la quale crea i cubi di dati OLAP e li mette a disposizione dell'applicazione web, dove sono presenti le reportistiche per gli utenti di business (nell'applicazione web, a seconda del tipo di dato da analizzare, possono esistere anche reportistiche che si basano su pacchetti transazionali, cioè che leggono direttamente dal DB).

Rispetto all'architettura che abbiamo descritto alla fine del secondo capitolo, notiamo che qui manca il repository atto a contenere i dati e le misure per la gestione della qualità. Il nostro obiettivo è quindi quello di costruirne uno prendendo in considerazione le entità principali che possono essere di interesse per il modello di impresa di una multinazionale farmaceutica.

3.2. Creazione dei meta modelli delle entità

Secondo il modello di business dell'azienda, possiamo identificare alcune delle entità più importanti presenti nel DWH; quelle che abbiamo deciso di prendere in considerazione sono le seguenti:

- Forza Vendita (FV)
- Medici visitati dagli Informatori Scientifici del Farmaco (ISF)
- Vendite agli ospedali
- Numero di campioni elargiti ai medici
- Costi di gestione

Di seguito descriviamo ad una ad una le entità elencate e creiamo un meta modello grafico per capire le relazioni delle diverse entità con il mondo reale.

3.2.1. Forza Vendita

La forza vendita è suddivisa in diverse Business Unit (BU), cui fa capo il responsabile BU, le quali sono a loro volta suddivise in linee di promozione (a seconda dei farmaci che vengono promozionati dagli informatori), Ogni linea di promozione, che copre tutto il territorio nazionale, è affidata a diverse persone che coprono ruoli diversi a seconda dell'area geografica di cui sono responsabili; queste persone sono i Regional Sales Manager, gli Area Manager e gli Informatori Scientifici del Farmaco.

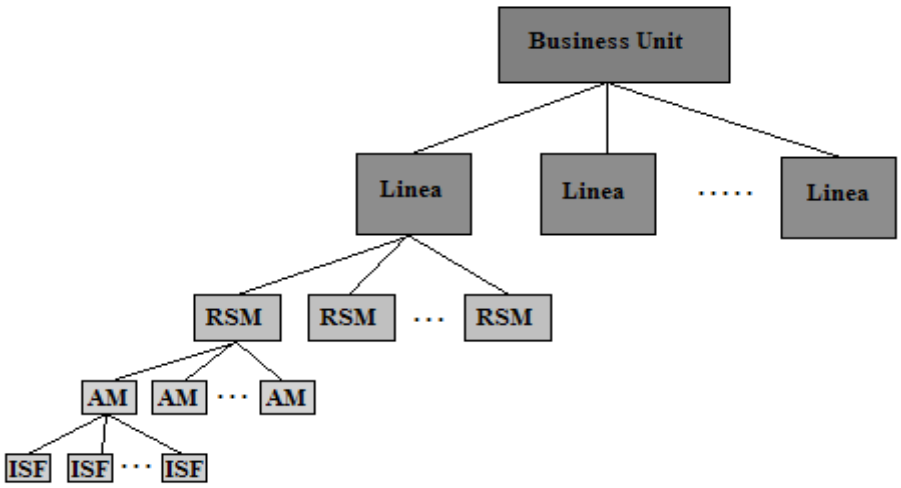


Figura 16 - Gerarchia della Forza Vendita

La forza vendita non viene distribuita sul territorio italiano tramite una distribuzione geografica (comuni, città, provincia) bensì attraverso una distribuzione *territoriale,* composta da due principali oggetti: il **microbrick** e il **brick**. Il microbrick corrisponde ad un estensione geografica pari a circa tre CAP, mentre un brick può contenere un numero variabile di microbrick. L'associazione tra informatore e brick viene decisa dai responsabili di ogni linea che strutturano gli ISF sui 721 brick che coprono il territorio nazionale decidendo se un solo ISF cpre uno o più brick, oppure se devono essere condivisi da più informatori. Gli informatori, inoltre, quando vengono assegnati a un brick, ereditano tutti i microbrick che stanno al livello sotto del brick stesso.

La suddivisione territoriale è importante in quanto identifica precisamente in che area l'informatore del farmaco deve svolgere il suo lavoro; inoltre su questa struttura è possibile associare dei dati di

vendita in maniera tale da misurare le performance degli stessi ISF. In particolare esistono alcune società specializzate nella vendita di questi dati (per esempio la IMS Health S.P.A) che offrono numerosi servizi per i più diversi tipi di analisi; nel nostro caso ne prendiamo in considerazione solo due:

- Servizio territoriale italiano contenente i dati di acquisto dei farmaci da parte delle farmacie italiane a livello territoriale
- Servizio contenente i dati di vendita delle farmacie in relazione ai farmaci rimborsati dal Servizio Sanitario Nazionale a livello territoriale

Questi dati possono essere acquistati in diversi modi; si possono scegliere i prodotti di interesse, fino al dettaglio della confezione, divisi per mercato di interesse (Oncologia, Primary Care, Ophtalmics, ecc…ecc…), acquistare i dati dei prodotti competitor per effettuare analisi di mercato e altre diverse opzioni.

Come già indicato in precedenza, i dati servono non solo per sapere come l'azienda si posiziona rispetto a un determinato mercato, ma anche per valutare le performance di ogni informatore scientifico del farmaco e, di conseguenza dei responsabili di riferimento (Manager dell'area o della regione); il processo di valutazione di un ISF viene chiamato "*processo di incentivazione*" e a seconda di quanto un ISF performa bene, tanto più avrà un *incentivo* come bonus in busta paga. Pertanto è evidente che sia i dati relativi alla struttura della forza vendita che quelli dei dati di vendita devono essere caricati correttamente in un DWH onde evitare problemi di non corretta distribuzione di premi in denaro sulle persone della FV.

Cerchiamo, ora, di analizzare l'entità forza vendite secondo le tre prospettive (concettuale,logico e fisico) descritti nei capitoli precedenti:

- **Prospettiva concettuale**

  Secondo la prospettiva concettuale, la forza vendita è la gerarchia delle persone che si occupano della promozione dei farmaci dell'azienda. La forza vendita ha quindi il compito, attraverso il lavoro di promozione, di far aumentare le vendite dei farmaci dell'azienda rispetto ai competitor.

- **Prospettiva Logica**

  Secondo la prospettiva logica, la forza vendite è in relazione ad altre entità che sono:

  a) La struttura territoriale (brick e microbrick) su cui è divisa la forza vendita
  b) Le informazioni sulle vendite/acquisti dei grossisti/farmacie
  c) Obiettivi di vendita per il pagamento dei bonus

- **Prospettiva fisica**

  Secondo la prospettiva fisica, i dati della struttura della forza vendita devono essere immagazzinati in una tabella che ne garantisca la gerarchia; i legami con le entità descritte nella prospettiva logica possono avvenire per mezzo di viste o viste materializzate a seconda del tipo di codice e del numero di record che vengono ricavati (si usa una vista se le interrogazioni sono rapide, se invece la mole di dati è molto ampia è meglio utilizzare una vista materializzata, prendendo atto del tempo necessario per il refresh di quest'ultima).

Dopo aver effettuato l'analisi dell'entità reale attraverso le tre prospettive del nostro modello, andiamo ora a creare una rappresentazione grafica di quello che può essere il nostro data mart della qualità per la *forza vendita*.

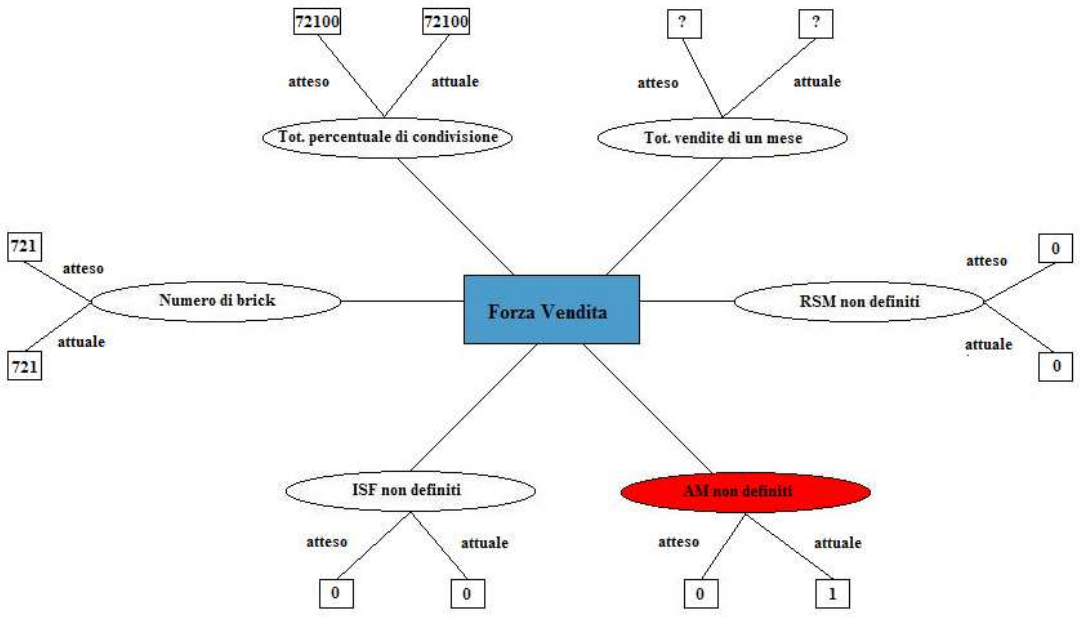


Figura 17 - data mart della qualità per l'entità forza vendita

Il metodo grafico serve per controllare in maniera rapida se ci sono problemi di qualità. Gli ellissi rossi indicano dove sono necessari controlli di qualità o dove gli utenti si devono aspettare bassa qualità. Analizzando in dettaglio il grafico possiamo valutare tutti le relazioni che possono portare a una bassa qualità, con la corrispondenza tra valore atteso e valore attuale presente nel DWH.

Da sinistra verso destra vediamo che sia il numero di brick della struttura della forza vendita che il totale della percentuale di condivisione sono corretti (721 brick compongono l'Italia secondo la suddivisione territoriale; ogni brick deve essere condiviso al 100%, da qui il valore 72100). Per quanto riguarda la relazione con le vendite totali di un mese, questa è stata inserita con un punto di domanda, in quanto il valore varia da mese a mese; il controllo sta nel verificare se il totale delle vendite non ancora assegnato alla struttura corrisponde esattamente a quello che viene allocato sulla struttura della forza vendita (verificando nel nostro DWH troviamo che la corrispondenza è sempre corretta).

Verifichiamo ora la corretta struttura della forza vendita assicurandoci che non esistano figure "non definite", cioè che non ci siano ISF, AM o RSM che non vengono riconosciuti dal sistema. In questo caso, come si nota dal grafico, abbiamo trovato un caso in cui esiste un AM che non è definito; la causa è dovuta al fatto che questa persona è stata cancellata nell'anagrafica degli impiegati dell'azienda. Grazie al controllo di qualità effettuato, è stato possibile segnalare l'anomalia alle persone di riferimento dell'azienda ed intervenire il prima possibile onde evitare di consegnare dati sbagliati alle persone del business.

Sottolineiamo, inoltre, che i controlli indicati nel grafico, devono essere ripetuti per tutte le linee di incentivazione del farmaco (si veda la figura 16).

3.2.2. Medici visitati dagli informatori scientifici del farmaco (ISF)

Gli informatori scientifici del farmaco promuovono i propri prodotti, relativi alla business unit di competenza, facendo visita ai medici ospedalieri e ai medici di base. Pertanto, possiamo dire, con buona approssimazione, che il numero di visite effettuato dagli ISF può essere considerato quanto essi lavorino effettivamente. Per questa entità è quindi molto importante avere dei numeri corretti

da pubblicare sull'applicazione di business intelligence, onde evitare richiami alle persone della forza vendita laddove non sia necessario.

Ovviamente gli informatori non possono andare casualmente da un medico o da un altro a proporre i propri prodotti, soprattutto se parliamo di prodotti strettamente specialistici (ad esempio quelli oftalmici o quelli oncologici). A ciascun informatore viene assegnato uno *schedario*, cioè un gruppo di medici che deve essere visitato con determinata frequenza a seconda dell'importanza strategica che ha per l'azienda.

Parlando di frequenza con la quale un medico o un gruppo di medici (ad esempio tutti i medici in schedario) viene visitato, dobbiamo introdurre un'altra entità, che è quella delle *attività*. Se volessimo calcolare un KPI che ci dice qual è la frequenza di visita di un medico, dovremo prendere il totale delle visite fatte su un determinato medico, rispetto ai giorni lavorativi in cui l'informatore ha effettuato visite (escludendo quindi possibili giorni di ferie/malattia o altre attività ad esempo riunioni di settore, ecc…ecc…)

Le visite vengono inserite direttamente dall'informatore scientifico nel sistema CRM; solitamente il foglio delle attività e delle visite del mese precedente può essere compilato fino a i primi giorni del mese successivo (ad esempio le visite e le attività di marzo le posso confermare sul sitema CRM fino al 7 o 8 di aprile). Nel caso in cui gli ISF non possano per qualsiasi motivo completare le informazioni del proprio lavoro, viene inviata una richiesta alle persone della sede della società che hanno i privilegi adeguati sul sistema per poter inserire informazioni nel passato.

Vediamo ora in dettaglio questa entità nelle tre prospettive indicate dalla metodologia DWQ.

- **Prospettiva concettuale**

  Secondo la prospettiva concettuale, la visite effettuate dagli ISF presso i medici descrizione quanto viene incentivato un determinato prodotto di una determinata linea; serve anche per dare un'idea della quantità di lavoro erogata dall'ISF.

- **Prospettiva Logica**

  Secondo la prospettiva logica, la visite effettuate dagli informatori scientifici del farmaco sono in relazione ad altre entità che sono:

a) L'anagrafica dei medici visitati
b) Il proprio schedario di medici da visitare
c) Il numero di giorni lavorati

- **Prospettiva fisica**

  Secondo la prospettiva fisica, i dati riguardanti i medici visitati e le visite effettuate dagli ISF devono essere inseriti entro 10 dalla chiusura del mese precedente. I processi del DWH, tuttavia, per non appesantire le macchine vanno a coprire ogni giorno uno storico di 4 mesi per recuperare le informazioni su questa entità. Informazioni che vengono modificate ma con data di validità superiore ai 120 giorni, non vengono caricate in DWH a meno di richieste puntuali.

Andiamo ora a creare il nostro data mart grafico della qualità per l'entità *Medici visitati dagli ISF*.

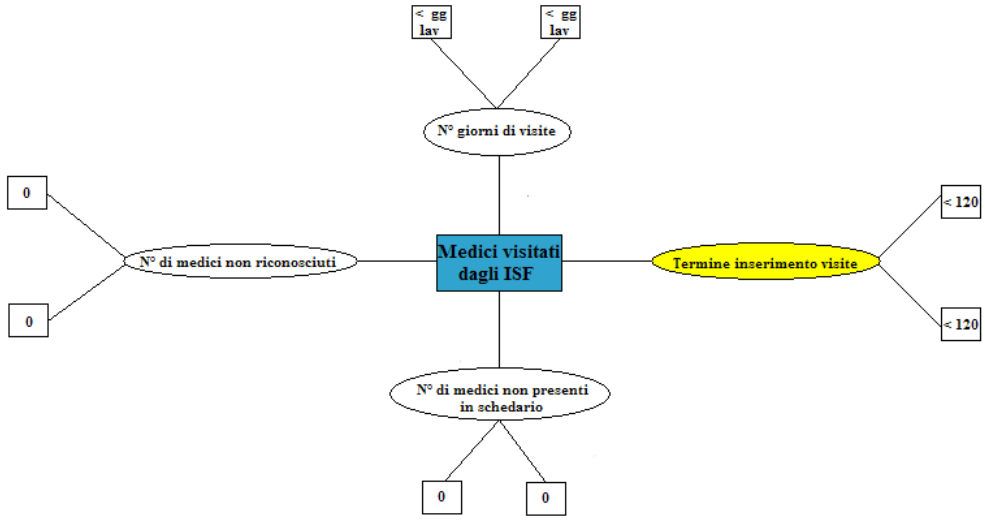


Figura 18 - data mart della qualità dell'entità medici visitati dagli ISF

Come possiamo vedere nel nostro DWH i dati sono già puliti in quanto non si segnalano, al momento dello studio, anomalie sui dati. L'unico ovale in giallo ricorda che c'è un vincolo (non controllabile direttamente sul DWH) di 120 giorni per l'inserimento delle visite sul sistema CRM.

3.2.3. Vendite agli ospedali

Come indicato nei paragrafi precedenti, le società farmaceutiche acquistano informazioni riguardanti alle vendite dei grossisti e/o delle farmacie sul territorio nazionale. Esistono altri dati di vendita chiamate "vendite interne", che sono quelle che l'azienda svolge direttamente presso gli ospedali o i grossisti e che vengono registrate all'interno del sistema ERP. Nel nostro caso, prenderemo in considerazione solamente le vendite legate agli ospedali.

Da un punto di vista concettuale, le vendite interne rappresentano il fatturato dell'azienda. Pertanto appare palese l'importanza della correttezza di questi dati per le persone del business. Inoltre, le sole vendite ospedaliere, si possono riagganciare al discorso di premi di produzione e incentivi per gli ISF, ma questo lo vedremo in seguito parlando delle relazioni delle vendite con le altre entità in gioco.

Descrivendo più specificamente questa entità, possiamo dire che questa si può misurare attraverso te tipologie di dato:

- *Quantità*: rappresenta il numero di confezioni di farmaco vendute agli ospedali
- *Valore*: rappresenta l'ammontare, in euro, dei farmaci venduti agli ospedali
- *Equivalenti*: misura simile alla quantità, rappresenta il numero di farmaci venduti agli ospedali facendo distinzione tra le singole confezioni; questo significa le vendite degli stessi prodotti ma in confezioni con un dosaggi diversi verranno calcolate in maniera diversa (se ad esempio il farmaco x nella confezione 1 contiene 300 mg di un reagente mentre nella confezione 2 contiene 150 mg dello stesso reagente, allora le vendite del farmaco x saranno calcolate come : *2x(vendite conf1) + vendite conf2*)

Considerando le vendite interne come il fatturato dell'azienda decidiamo di prendere in considerazione il calcolo tramite la misura *Valore*. Inoltre, se avessimo utilizzato la misura quantità o equivalenti, avremmo dovuto togliere dal valore calcolato i possibili sconti merce, che vengono si calcolati nel sistema come quantità vendute, in quanto correlati da fattura, ma che in realtà non vengono fatte pagare al cliente (sono quindi farmaci usati come "promozione").

Passiamo ora allo studio di questa entità secondo la prospettiva logica. Per prima cosa possiamo notare che le vendite ospedaliere sono legate a due entità reali che sono presenti nello stesso sistema ERP; i prodotti farmaceutici e gli ospedali. Ai fini della reportistica che viene visualizzata dalle persone del business, è importante avere un'anagrafica completa e corretta di entrambe le entità; se così non fosse si correrebbe il rischio di avere dei report incompleti o non corretti. La reportistica risulterà incompleta se esistono fatture relative a un prodotto o a un ospedale non censito nel sistema ERP, non corretta se esistono fatture relative a prodotti od ospedali inseriti scorrettamente nel sistema ERP e caricati nel DWH.

Un altro strumento importante è quello che lega gli ospedali, sui quali andiamo a valutare il valore delle vendite, agli informatori scientifici; infatti, anche il valore delle vendite agli ospedali danno un'evidenza della bontà del lavoro degli ISF. Inoltre, è possibile calcolare obiettivi di vendita e incentivi pianificando una determinata distribuzione della FF sugli ospedali del territorio nazionale. Diventa importante, in questo contesto, essere certi di aver legato tutti gli ospedali agli uomini, onde evitare che determinate vendite, frutto del lavoro dell'ISF, non vengano conteggiate (questo è un caso che può capitare se viene inserito in anagrafica un nuovo ospedali e non viene assegnato in tempo breve).

Esaminando le vendite ospedaliere da una prospettiva fisica, possiamo valutare la velocità di interrogazione che è presente sulle viste del DWH. Poiché questi dati vengono molto richiesti per i motivi sopra indicati (controllo vendite, obiettivi e incentivi) è necessario riuscire ad ottenere i dati in forma corretta nel minor tempo possibile. Inoltre, il sistema ERP contiene i dati aggiornati in tempo reale e ogni notte gli ETL portano le informazioni relative al giorno prima sul DWH (ad esempio se oggi è il 7 di ottobre, dovrò avere i dati aggiornati fino al 6). Poiché non sempre gli ETL vanno a buon fine, o non sempre il sistema ERP mette a disposizione un file corretto, il controllo dell'ultima data di vendita diventa importante per capire istantaneamente se ci sono stati dei problemi nei caricamenti schedulati e se si necessita di fare dei caricamenti manuali.

Riassumiamo, quindi, l'analisi dell'entità nelle tre prospettive:

- **Prospettiva concettuale**

  Secondo la prospettiva concettuale, la vendite interne, delle quali abbiamo analizzato il sottogruppo di quelle ospedaliere, rappresentano il fatturato dell'azienda.

- **Prospettiva Logica**

  Secondo la prospettiva logica, le vendite ospedaliere sono in relazione ad altre entità che sono:

  a) Le anagrafiche dei prodotti e degli ospedali presenti nel sistema ERP.
  b) La Field Force (FF)
  c) Gli obiettivi di vendita per il pagamento dei bonus

- **Prospettiva fisica**

  Secondo la prospettiva fisica, le vendite ospedaliere devono:

  a) essere aggiornati sempre con l'ultima data di vendita del sistema ERP (solitamente data di sistema -1)
  b) immagazzinati in oggetti che ne permettano una rapida interrogazione senza consumare troppe risorse di sistema e senza grossi tempi di aggiornamento per non impattare i processi che già vengono eseguiti giornalmente.

Dopo aver effettuato l'analisi dell'entità reale attraverso le tre prospettive del nostro modello, andiamo ora a creare una rappresentazione grafica di quello che può essere il nostro data mart della qualità per le *vendite ospedaliere*.

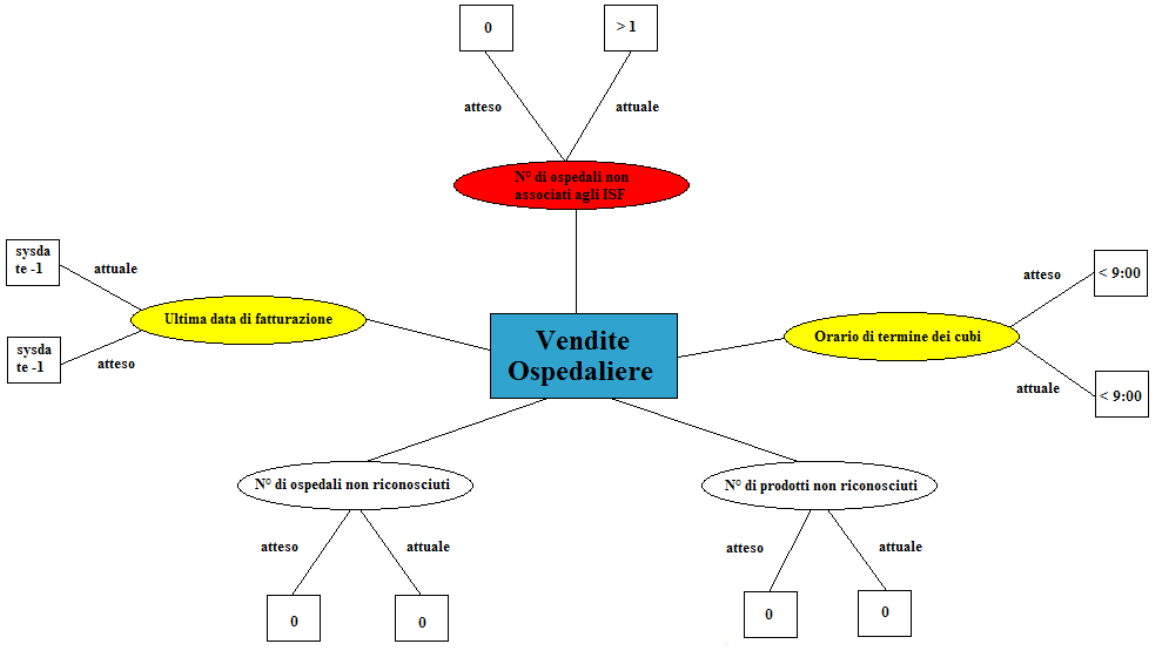


Figura 19 - data mart della qualità per l'entità vendite ospedaliere

Come si può notare dalla figura, i due ovali che rappresentano un controllo di qualità dal punto di vista fisco del DWH sono in giallo. Questo perché più che un controllo vero e proprio di qualità del dato, danno l'idea di alcuni controlli che vanno fatti per identificare un corretto funzionamento del data base su cui sono presenti i dati. In particolare, il controllo sull'ultima data di fatturazione, non solo indica se ci sono stati problemi sul DWH, bensì se ci sono ritardi da parte del sistema ERP per l'invio delle informazioni necessarie.

L'unico controllo di qualità che risulta essere negativo, e di conseguenza segnalato con un ovale rosso, è quello degli ospedali non associati ad alcun ISF. L'associazione ISF-Ospedale viene eseguita tramite portale online dalle persone di sede di ogni linea, che hanno la responsabilità di mantenere aggiornati questi legami. Poiché ogni giorno può entrare in anagrafica un nuovo ospedale al quale sono associate vendite, bisognerebbe aggiornare costantemente il portale, in maniera tale da mantenere una reportistica territoriale delle vendite completa. Per evitare che alla chiusura fiscale del mese ci siano situazioni di questo tipo, è stato creato un processo che 4 gg prima della scadenza manda una mail a ogni responsabile di linea con l'indicazione di quanti e quali ospedali risultano ancora non essere associati. Dall'introduzione di questo processo sono stati annullati i ritardi alla chiusura fiscale del mese dovuti alla mancanza di vendite legate.

3.2.4. Saggi consegnati ai medici

Durante l'attività di visita degli informatori scientifici del farmaco ai rispettivi medici, solitamente, viene accompagnato il rilascio di un numero di campioni di un determinato farmaco.

Un campione non è altro che un farmaco che deve essere graficamente identico alla confezione più piccola messa in commercio; il suo contenuto può essere, e solitamente è, inferiore, in numero di unità posologiche o in volume, a quello della confezione in commercio, purché risulti terapeuticamente idoneo.

Dal 2007 lo scarico dei saggi è stato regolamentato in maniera molto rigida tramite una nuova normativa europea (D.Lgs. 30.12.92 n. 541 in attuazione della direttiva 92/28 CEE), prevedendo gravi ammende per le società che non rispettano i limiti imposti. In particolare, la normativa dice che:

1. I campioni gratuiti di un medicinale per uso umano possono essere rimessi solo ai medici autorizzati a prescriverlo e *devono essere consegnati soltanto per il tramite di informatori scientifici*
2. I campioni *non possono essere consegnati* senza una richiesta scritta, recante data, timbro e firma del destinatario.
3. Gli informatori scientifici possono consegnare a ciascun sanitario *due campioni a visita per ogni dosaggio o forma farmaceutica* di un medicinale esclusivamente *nei diciotto mesi successivi* alla data di prima commercializzazione del prodotto ed entro il limite massimo di otto campioni annui per ogni dosaggio o forma
4. Fermo restando il disposto del comma 2, *gli informatori scientifici possono inoltre consegnare al medico non più di quattro campioni a visita, entro il limite massimo di dieci campioni annui,* scelti nell'ambito del listino aziendale dei medicinali in commercio da più di diciotto mesi.
5. *I limiti quantitativi dei commi 3 e 4 si applicano anche ai medicinali vendibili al pubblico in farmacia non dispensati con onere a carico del Servizio sanitario nazionale.*

Riassumendo possiamo dire che, secondo la prospettiva concettuale, i saggi sono dei campioni di farmaci il cui scarico è regolamentato per legge secondo determinate restrizioni.

Da un punto di vista logico, andiamo a individuare quali sono le relazioni che legano i saggi ad altre entità. Per prima cosa abbiamo un legame con i prodotti; i saggi consegnati devono essere presenti in anagrafica prodotti in maniera tale che il sistema li riconosca e non perda record preziosi. In seconda battuta i saggi sono legati all'ISF che li ha consegnati e al medico cui sono stati consegnati. Anche in questo caso, è molto importante che il medico sia presente nell'anagrafica dello schedario dell'ISF, altrimenti il sistema non sa a chi attribuire il ricevimento del farmaco (e visto che ci sono dei limiti di legge per ogni medico non riconoscere dei moduli sarebbe molto rischioso per l'azienda).

Andiamo ora a creare il nostro data mart grafico della qualità per l'entità *saggi consegnati dagli informatori scientifici*.

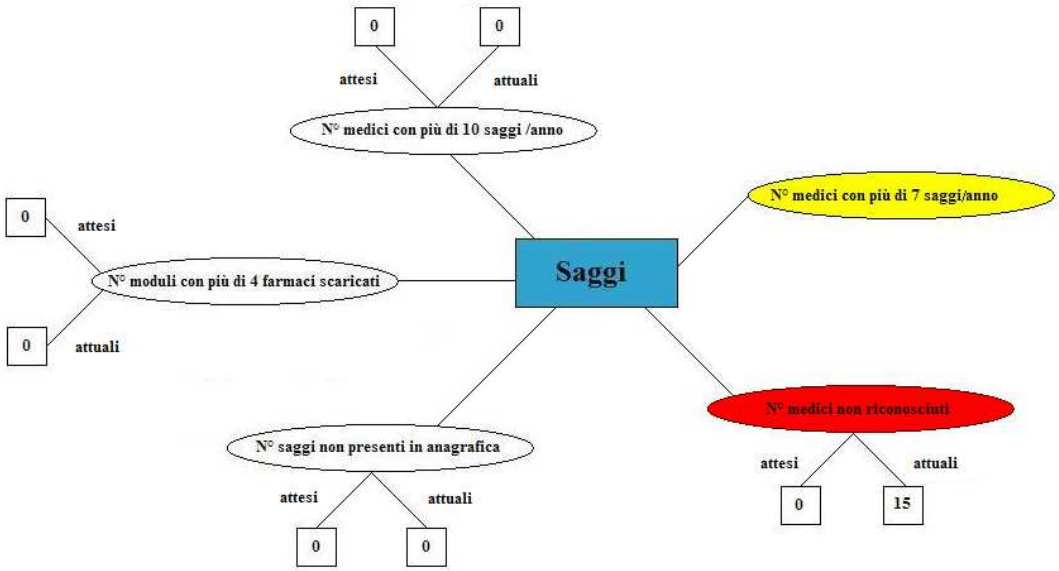


Figura 20 - data mart della qualità per l'entità saggi consegnati dagli informatori scientifici

Come possiamo notare dal grafico, l'unica relazione che sembra non essere soddisfatta è quella del riconoscimento del medico a cui sono stati consegnati i saggi; questo perché il nome del medico viene scritto manualmente dall'ISF e il sistema può non riconoscere la scrittura a mano (tramite penna). Per migliorare questo aspetto della qualità, è stata realizzata una procedura PL/SQL che attraverso degli algoritmi di somiglianza (tipo quelli di Accard o di Jaro-Winkler), verifica quanto riconosciuto della scrittura dell'informatore, con i propri medici in schedario, partendo da quelli più strategici. Nel caso in cui non venga riscontrato nessun match che superi una determinata soglia di assomiglianza (ad esempio l'80%) viene inviata una mail attraverso una procedura java nella quale viene indicato all'ISF di accedere al portale web ed inserire nuovamente, stavolta tramite pc, il nome del medico.

L'ovale giallo, invece, indica una relazione alla quale bisogna dare particolarmente attenzione. Nel nostro caso, visto che i limiti di legge impongono di non dare più di 10 campioni ad un determinato medico, abbiamo deciso di creare un "warning". Poiché un medico può essere visitato da più informatori (quindi essere presente in entrambi i schedari) e visto che ogni informatore sa solo quanti saggi ha consegnato egli stesso al medico, abbiamo implementato un sistema che, quando il medico arriva vicino alla soglia dei 10 campioni, invii una mail agli informatori associati al medico

stesso. In questo modo diminuiscono le probabilità di sforare il limite e di incorrere in una multa per la società.

Questo modo di gestire i saggi degli informatori, ha aperto le porte per nuovi spunti su come far diminuire i costi dell'azienda e, di conseguenza, migliorare la qualità del processo. Ad esempio, poiché la società invia un determinato numero di campioni ogni 4 mesi, tracciando i saggi consegnati si può capire quanti sono i saggi che rimangono in giacenza presso l'informatore, saggi che poi devono essere rispediti alla società per lo smaltimento. Tramite alcuni studi statistici di quanti sono i saggi che vengono distribuiti, si può calibrare il numero di saggi consegnati inmodo tale da abbattere i costi sia di invio che di smaltimento.

Un altro aspetto ancora di miglioramento potrebbe essere quello di personalizzare le spedizioni a seconda dei campioni che vengono consegnati; se da un lato questo porterebbe a una drastica diminuzione degli smaltimenti dei farmaci, dall'altro i costi di spedizione di pacchetti diversificati aumenterebbero esponenzialmente. Tramite il tracciamento, in qualunque caso, si potrà valutare quale sia la soluzione più vantaggiosa.

3.2.5. Costi di gestione

Fino ad ora abbiamo trattato entità che descrivevano la quantità di lavoro e la quantità di denaro che l'azienda riceve; ci occuperemo ora, invece, dell'altra faccia della medaglia che è rappresentata dai costi di gestione. I costi di gestione vengono utilizzati nell'ambito della business intelligence per il calcolo dei Profit & Loss (P&L) delle varie aree. Nel nostro caso, questi costi vengono distribuiti sulla struttura territoriale della Field Force e vengono suddivisi a seconda del prodotto che viene venduto dalla società.

Possiamo descrivere 3 tipologie di costi di cui vengono caricati le informazioni nel DWH della nostra azienda: i costi della field force (che non sono altro che gli stipendi delle persone che lavorano sulla linea di promozione), i costi dei samples (cioè il costo dei campioni che vengono rilasciati ai medici) e un'altra categoria di costo che comprende tutto il resto (ad esempio i costi dovuti alle trasferte, alle presentazioni ecc…ecc…).

I costi rappresentano, quindi, la quantità di denaro che l'azienda investe per la promozione e la vendita dei propri prodotti; abbinando queste informazioni a quelle delle vendite è possibile identificare l'utile totale dell'azienda e il ricavo netto per ogni singola porzione territoriale degli informatori. Queste informazioni saranno utilizzate successivamente dagli utenti di business per decidere la strategia di marketing migliore per far recuperare le zone con meno redditività.

I costi di gestione sono contenuti nel sistema ERP e vengono caricati sul DWH tramite ETL; questi costi sono spesso associati o a dei Profit Center o a degli Internal order che servono per discriminare le categorie di costi e le linee di produzione alle quali vanno attribuiti. I costi, una volta caricati nel DWH, vengono attribuiti alle linee e alla forza vendita (i costi di FF) e vengono visualizzati all'interno della reportistica messa a disposizione degli utenti di business.

Vediamo ora questa entità secondo le prospettive concettuale, logica e fisica:

- **Prospettiva concettuale**

  Secondo la prospettiva concettuale, i costi di gestione rappresentano quanto l'azienda investe per la promozione e la distribuzione dei farmaci prodotti.

- **Prospettiva Logica**

  Secondo la prospettiva logica, i costi di gestione sono in relazione ad altre entità che sono:

  a) I profit center mappati nel DWH
  b) La Field Force (FF)
  c) Gli internal order mappati nel DWH
  d) L'anagrafica Prodotti

- **Prospettiva fisica**

  Secondo la prospettiva fisica non ci sono particolari impatti su questa entità, in quanto i caricamenti vengono fatti su base mensile e i record caricati non sono molti (nell'ordine delle migliaia). L'unico impatto è dovuto al fatto che il caricamento di questi dall'estrazione del sistema ERP alla presentazione sull'applicazione online deve avvenire in 4 giorni lavorativi.

Andiamo ora a creare il nostro data mart grafico della qualità per l'entità *Costi di gestione*.

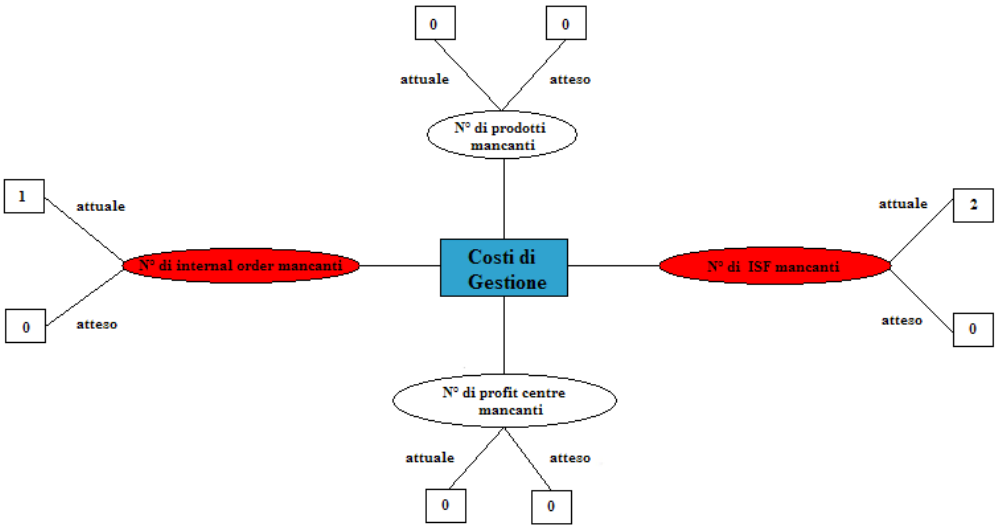


Figura 21 - data mart della qualità per l'entità Costi di gestione

Come possiamo notare, ci sono due punti di attenzione evidenziati dagli ellissi in rosso; il primo riguarda gli internal order presenti nel DWH , il secondo riguarda la corrispondenza con la struttura della FF.

Per quanto riguarda gli internal order mancanti, significa che nel file di costi è apparso un internal order che non è ancora stato mappato all'interno del DWH. Bisognerà, pertanto, avvisare gli utenti di competenza affinché inseriscano questa informazione; fino a quando questa informazione mancherà, i costi relativi all'Internal order non saranno caricati e nemmeno visibili, falsando così il totale dei costi dell'azienda sulla reportistica.

Per quanti riguarda la FF, è possibile che si siano cambiamenti all'interno delle linee come l'aggiunta o l'eliminazione di alcuni ISF o uno scambio tra linee. Il sistema che si occupa di gestire i costi ha, tuttavia, un legame stretto tra un determinato profit center e un ISF. Se la struttura cambia e non vengono riviste le associazioni, la reportistica estrarrà dei dati non consistenti con quelli caricati sul database del DWH. Diventa, quindi, importante, a valle di ogni modifica della struttura che venga eseguita l'operazione cosiddetta di "restatement" che consiste nel cancellare i costi fino ad ora allocati su una struttura e partizionarli nuovamente sulla nuova struttura della forza vendita.

3.3.Realizzazione del data mart della qualità sul DWH

Vediamo ora come passare dalla visualizzazione grafica dei data mart delle singole entità descritte nel paragrafo precedente alla loro realizzazione vera e propria. Per prima cosa dobbiamo decidere se mantenere il controllo qualità nello stesso schema dove si trovano i dati o se vogliamo creare uno schema a parte che contenga solo le tabelle con le misure di qualità. La seconda soluzione è sicuramente la più pulita, dobbiamo essere sicuri, però, che tutti gli oggetti che andiamo a leggere dallo schema dei dati abbiamo i permessi corretti (altrimenti le interrogazioni andranno in errore).

Creerò, successivamente, una tabella nella quale inserire le nostre misure di qualità; la prima colonna conterrà il nome dell'entità, la seconda colonna il nome del controllo e la terza il valore atteso. Si può decidere di aggiungere anche una colonna con lo stato attivo e disattivo e una colonna con il timestamp di inserimento del valore atteso, in maniera tale da mantenere lo storico delle misure di qualità che si vogliono adottare.

Bisogna decidere ora in che modo rendere fruibili i dati relativi alla qualità, in particolar modo, quali sono i controlli che hanno una valore attuale diverso da quello atteso (che indica, quindi, della bassa qualità). Possiamo decidere di visualizzare ogni giorno il valore calcolato tramite interrogazioni, storicizzare i controlli che non risultano essere positivi con la relativa data, in modo tale da poter fare delle analisi storiche sui dati, oppure creare un processo che mandi via mail i risultati a dei gruppi di competenza con una cadenza concordata.

La soluzione che realizzerò, sarà quella di creare delle procedure che popolino una o più tabelle (a seconda del numero di dati da inserire) che saranno lette da una vista che unirà i valori attesi con quelli riscontrati. La vista verrà poi letta a sua volta da un'applicazione web nella quale si potrà scegliere l'entità di interesse e i periodi temporali per l'analisi della quantità delle casistiche di controlli negativi. Gli utenti di business potranno vedere così on-line ogni giorno i dati relativi alla qualità; ulteriore sviluppo sarà quello di filtrare, a seconda dell'utenza che si collega, solo le entità di proprio interesse, oppure filtrare i dati relativi alle varie entità per ogni singola business unit. Il risultato sarà una pagina html dalla quale si visualizzerà la numerosità dei controlli negativi per una determinata entità suddivisi per nome del controllo, dove cliccando sul numero si eseguirà un report

di dettaglio con tutte le date in cui si è riscontrata una bassa qualità. Infine, entrambi i report saranno esportabili in file excel.

Per prima cosa creiamo un utente ad hoc per il nostro data mart con il seguente script:

```
Create user dwh_quality
Identified by qualita
Default tablespace ts_user
Temporary tablespace ts_temp;
```

Una volta creato il nuovo schema, dobbiamo abilitarlo alla lettura delle tabelle che ci servono per fare il controllo di qualità. Dai paragrafi sopra le entità che ci interessano sono le vendite interne, i medici in schedario, la struttura della forza vendita, i campioni consegnati e i costi di gestione, quindi:

```
grant select on table_sales to dwh_quality
grant select on table_customer_ff to dwh_quality
grant select on table_field_force to dwh_quality
grant select on table_samples to dwh_quality
grant select on table_expenses to dwh_quality
```

Ora che il nostro utente per la qualità ha i permessi per leggere i dati delle entità che ci interessano, andiamo a creare la tabella che conterrà le informazioni sui nostri alert; successivamente ci preoccuperemo di creare la procedura che popola i dati.

Per creare la tabella usiamo il seguente script:

```
create table misura_qualita
(Entita          varchar2(50)
 Controllo       varchar2(200)
 Valore_atteso   varchar2(10)
 Flag_attivo     varchar2(1)
 Time_stamp      date
);
```

e successivamente la popoliamo con i valori attesi delle nostre misure di qualità:

```
Insert into misura_qualita values ('forza vendita','ISF non definiti','0','Y','25/09/2010');
Insert into misura_qualita values ('forza vendita','AM non definiti','0','Y','25/09/2010');
Insert into misura_qualita values ('forza vendita','Numero di brick','721','Y','25/09/2010');
Insert into misura_qualita values ('forza vendita','RSM non definiti','0','Y','25/09/2010');
Insert into misura_qualita values ('forza vendita','Tot. Percentuale di condivisione','72100','Y','25/09/2010');
Insert into misura_qualita values ('medici visitati','N medici non presenti in schedario','0','Y','25/09/2010');
Insert into misura_qualita values ('medici visitati','N medici non riconosciuti','0','Y','25/09/2010');
Insert into misura_qualita values ('vendite agli ospedali','Numero ospedali non riconosciuti','0','Y','25/09/2010');
Insert into misura_qualita values ('vendite agli ospedali','Numero prodotti non riconosciuti','0','Y','25/09/2010');
Insert into misura_qualita values ('vendite agli ospedali','N ospedali non associati ad ISF','0','Y','25/09/2010');
Insert into misura_qualita values ('Numero di campioni','N medici non riconosciuti','0','Y','25/09/2010');
Insert into misura_qualita values ('Numero di campioni','N saggi non presenti in anagrafica','0','Y','25/09/2010');
Insert into misura_qualita values ('Numero di campioni','N moduli con + di 4 saggi','0','Y','25/09/2010');
Insert into misura_qualita values ('Numero di campioni','N medici con + di 10 saggi/anno','0','Y','25/09/2010');
Insert into misura_qualita values ('Costi di gestione','N profit center mancanti','0','        Y','25/09/2010');
Insert into misura_qualita values ('Costi di gestione','N internal order mancanti','0','Y','25/09/2010');
Insert into misura_qualita values ('Costi di gestione','N prodotti mancanti','0','Y','25/09/2010');
Insert into misura_qualita values ('Costi di gestione','N di ISF mancanti','0','Y','25/09/2010');
```

Ora che la tabella è pronta, dobbiamo ricavare i valori attuali, giorno per giorno, delle nostre misure di qualità. Come indicato precedentemente nel paragrafo, l'idea è quella costruire una nuova tabella da affiancare alla nostra *misura_qualita*, per poter estrarre i valori di interesse tramite una vista.

Creiamo allora la tabella *valori_attuali_qualita* col seguente script:

```
create table valori_attuali_qualita
(Entita            varchar2(50)
 Controllo         varchar2(200)
 Valore_attuale    varchar2(10)
 Giorno    date
);
```

I campi verranno aggiornati da una serie di script .sql che saranno inseriti in una sequence di un ETL, in modo tale da poter sfruttare le potenzialità del programma per avere un log tracciato e per poter schedulare in maniera semplice l'esecuzione degli script stessi (per il codice degli script vedi Appendice).

Ora che abbiamo le due tabelle base completate, dobbiamo costruire o un oggetto o le relazioni tra le due tabelle in modo tale che la reportistica online possa leggere i dati. Costruiamo, allora, due viste materializzate che contengono la prima il conteggio di quanti alert si sono verificati all'interno di un anno, la seconda che contiene il dettaglio del giorno in cui si è verificato.

```
Create materialized view conteggio_alert as
Select mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4) as anno, count(*) as totale
From   misura_qualita mq, valori_attuali_qualita vaq
Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='0'
And vaq.valore_attuale <> '0'
Group by mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4)
Union all
Select mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4) as anno, count(*) as totale
From   misura_qualita mq, valori_attuali_qualita vaq
Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='721'
And vaq.valore_attuale <> '721'
Group by mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4)
Union all
Select mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4) as anno, count(*) as totale
From   misura_qualita mq, valori_attuali_qualita vaq
Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='72100'
And vaq.valore_attuale <> '72100'
Group by mq.entita, mq.controllo, lpad(to_char(giorno,'yyyymmdd'),4);

Create materialized view dettaglio_alert as
Select mq.entita, mq.controllo, mq.valore_atteso, vaq.valore_attuale, vaq.giorno, lpad(to_char(giorno,'yyyymmdd'),4)
as anno
```

From misura_qualita mq, valori_attuali_qualita vaq

Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='0'

And vaq.valore_attuale <> '0'

Union all

Select mq.entita, mq.controllo, mq.valore_atteso, vaq.valore_attuale, vaq.giorno, lpad(to_char(giorno,'yyyymmdd'),4) as anno

From misura_qualita mq, valori_attuali_qualita vaq

Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='721'

And vaq.valore_attuale <> '721'

Union all

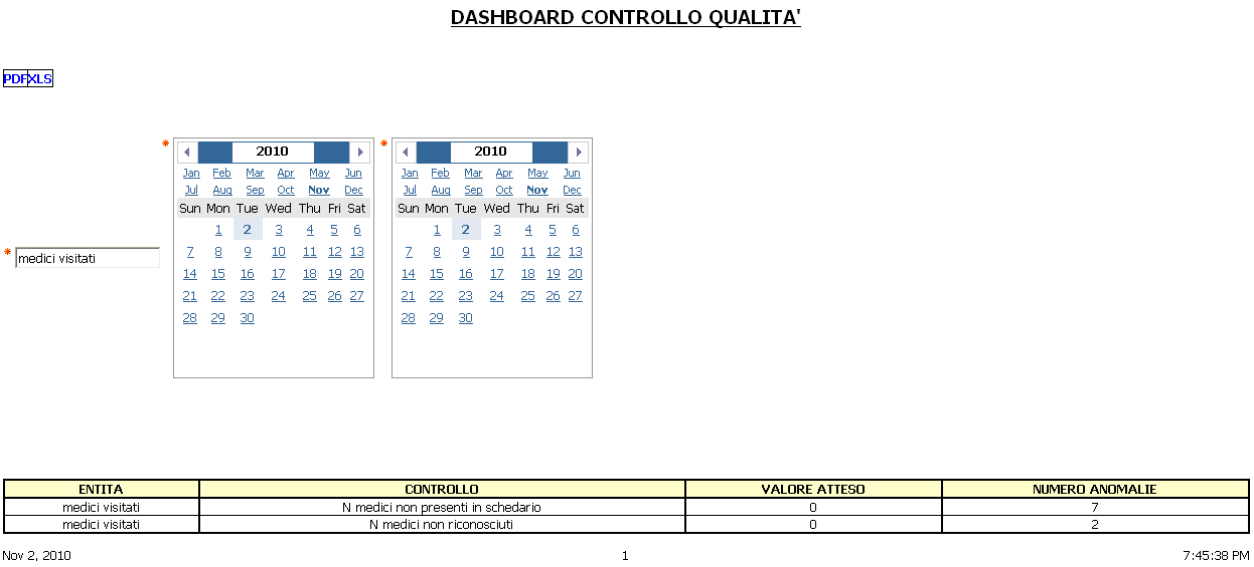
Select mq.entita, mq.controllo, mq.valore_atteso, vaq.valore_attuale, vaq.giorno, lpad(to_char(giorno,'yyyymmdd'),4) as anno

From misura_qualita mq, valori_attuali_qualita vaq

Where mq.flag_stato='Y' and mq.entita = vaq.entita and mq.controllo=vaq.controllo and mq.valore_atteso ='72100'

And vaq.valore_attuale <> '72100';

Una volta create queste due viste materializzate, andiamo a realizzare la reportistica che gli utenti di business avranno disponibile on-line; per far ciò utilizziamo la suite di business intelligence di IBM (Cognos). Il report mostra un prompt in cui è possibile scegliere l'entità da analizzare e due calendari per filtrare il periodo temporale dell'analisi. L'analisi è riportata in una tabella con l'indicazione dell'entità, del controllo, del valore atteso e del numero di anomalie riscontrate nel periodo selezionato. Inoltre è possibile, cliccando sulla tabellina in alto a sinistra, esportare l'analisi in formato .pdf o .xls.



DASHBOARD CONTROLLO QUALITA'

PDF XLS

medici visitati

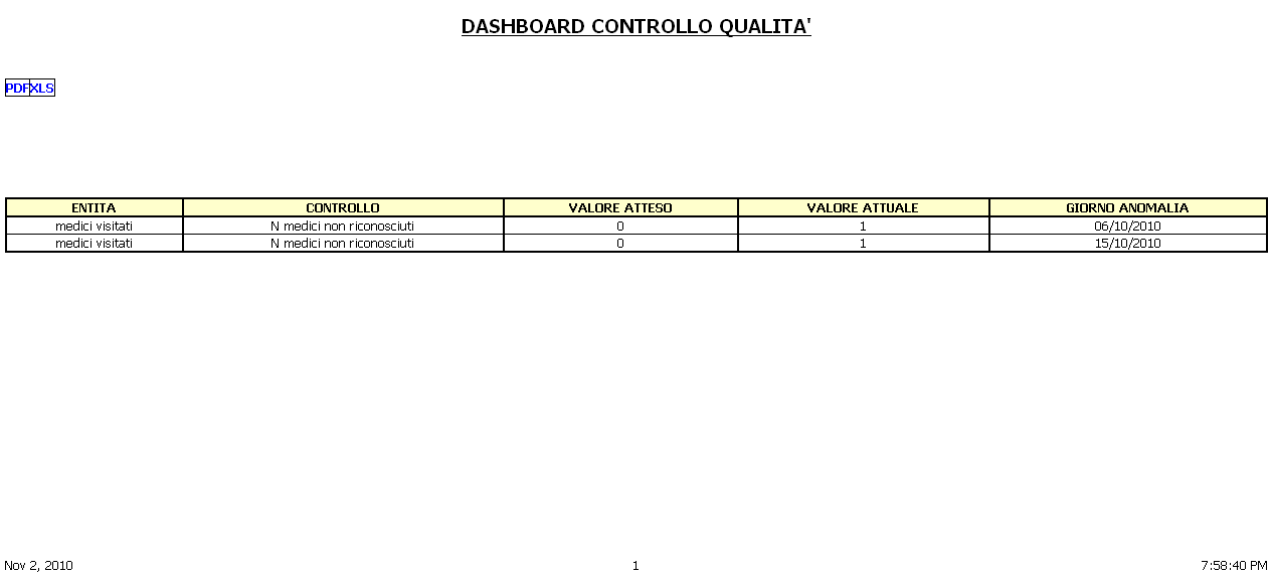
2010

ENTITA	CONTROLLO	VALORE ATTESO	NUMERO ANOMALIE
medici visitati	N medici non presenti in schedario	0	7
medici visitati	N medici non riconosciuti	0	2

Nov 2, 2010 1 7:45:38 PM

Figura 22 - Esempio reportistica controllo qualità

Se si clicca sul numero delle anomalie riscontrate si apre un report di dettaglio con le informazioni dei giorni in cui si è riscontrata l'anomalia e il valore riscontrato quel giorno



DASHBOARD CONTROLLO QUALITA'

PDF XLS

ENTITA	CONTROLLO	VALORE ATTESO	VALORE ATTUALE	GIORNO ANOMALIA
medici visitati	N medici non riconosciuti	0	1	06/10/2010
medici visitati	N medici non riconosciuti	0	1	15/10/2010

Nov 2, 2010 1 7:58:40 PM

Figura 23 - esempio report di dettaglio

anche in questo caso è possibile esportare il file in formato.xls o .pdf. Gli esempi mostrati sono molto semplici sia nella grafica che a livello di opzioni implementate; è possibile decidere di

implementare una multi selezione sulle entità, inserire dei grafici nel report di dettaglio o nel report generale oppure, senza lavorare a livello di reportistica, far si che ogni utente di business abbia dei permessi per vedere le entità filtrate sulle linee di competenza.

Conclusioni

La metodologia utilizzata dalle persone della DWQ, e riutilizzata per la realizzazione del data mart di qualità nell'azienda farmaceutica multinazionale presa come caso di studio, porta un grande vantaggio rispetto a tutte le altre soluzioni di data quality presenti in commercio; questo vantaggio è dato dal comprendere in maniera molto più approfondita il "business" (inteso come il funzionamento della realtà dell'azienda) di un determinato settore.

Le soluzioni principali di data quality presenti in commercio, servono per l'eliminazione di dati sporchi o di duplicazioni di dato; possiamo quindi affermare che la qualità intesa in queste soluzioni si restringe solo alla bontà del dato fine a se stesso. Sicuramente questi software sono molto importanti per una pulizia fisica del dato, soprattutto se stiamo parlando di anagrafiche (basti pensare alla necessità di un'azienda di telecomunicazioni di avere un'anagrafica clienti che non abbia duplicazioni con nomi e cognomi o indirizzi, solo per la diversa forma in cui si possono scrivere); quello che manca, e che invece viene ampliamente trattata nella metodologia del DWQ, è la visione di come funzionano i singoli flussi di un impresa, di un settore, e si "incastrano" nell'insieme.

Grazie a questa visione più ampia di come funziona davvero l'azienda, non solo si ha la possibilità di capire le motivazioni che sono dietro a determinate decisioni, ma si può responsabilizzare ogni singolo dipendente che utilizzi sistemi informativi affinché abbia coscienza che anche un inserimento di dati sbagliato o effettuato con dei ritardi rispetto alle scadenze può portare ad errate decisioni da parte degli utenti di business.

L'unico difetto che riscontro nell'utilizzo di questa metodologia è il fatto che non ci siano delle regole "rigide"; infatti, nonostante siano stati definiti i tre livelli a cui bisogna fare riferimento per l'analisi di qualità, è sempre la singola persona che prende atto delle entità dell'azienda e delle relazioni che coesistono tra di loro. Ciò fa si chi la metodologia, non sia "una scienza esatta", ma che sia una maschera utilizzata in modo oggettivo dalle persone che partecipano al progetto di qualità.

Questo fa si, come dicevo nell'introduzione, che ci sia bisogno di personale qualificato per svolgere questo tipo di analisi, personale che non sempre è presente in azienda (la business intelligence ha

preso piede negli ultimi 15 anni); anche nel nostro caso di studio, la parte di business intelligence relativa ai sistemi informativi (comprendiamo anche il data mar di qualità descritto nel capitolo precedente) è gestita da consulenti esterni all'azienda stessa.

Posso quindi affermare che al fine di avere un'ottima qualità nei dati di un DWH è necessario che le persone che lavorano utilizzando sistemi informativi (CRM, ERP, ecc… ecc…) abbiano un'ottima conoscenza del processo aziendale, di quali sono le entità reali dell'azienda stessa e quali sono le relazioni tra queste entità, in maniera tale da essere sensibilizzati a fare il proprio lavoro con cognizione di causa. Stesso discorso vale per gli utenti di business, che hanno il compito di prendere le decisioni e, quindi, di fissare obiettivi di qualità sul sistema, e per gli sviluppatori delle soluzioni di qualità sul DWH i quali, seppur partendo da requisiti formalizzati dagli utenti, devono avere una sensibilità tale da poter giudicare in maniera critica le richieste che vengono avanzate.

Bibliografia

(n.d.). Retrieved from DWQ – Foundations of Data Warehouse Quality: http://www.dbnet.ece.ntua.gr/~dwq/

Hinrichs, H. (2000). CLIQ – Intelligent Data Quality Management.

Jarke, M., & Vassiliou, Y. (1997). Data Warehouse Quality: A Review of the DWQ Project.

Jarke, M., Jeusfeld, M., Quix, C., & Vassiliadis, P. (1999). Architecture and Quality in Data Warehouse: an Extended Repository Approach.

Wang, R., & Strong, D. (1996). Beyond accuracy: What data quality means to data consumers.

Appendice

1. Script di inserimento valori nella tabella *valori_attuali_qualita* per controlli giornalieri

Qui di seguito verranno descritti gli script per il calcolo dei valori giornalieri per il popolamento della tabella *valori_attuali_qualita*.

Gli script vengono eseguiti in sequenza tramite un ETL che viene schedulato quotidianamente in tarda mattinata, per essere sicuri che il refresh giornaliero dei dati del DWH siano prima conclusi. All'interno dell'ETL sono presenti degli stage, chiamati transformer, nei quali vengono assegnati i valori per il coretto inserimento dei valori delle colonne *Entità*, *Controllo e Giorno*.

Gli script verranno presentati suddivisi per singola entità.

FORZA VENDITA

--ISF non definiti

select count(*) from (select distinct brick_code from struttura_terr where isf_no ='UD' and terr_status='Active')

--AM non definiti

select count(*) from (select distinct isf_no from struttura_terr where am_no ='UD' and terr_status='Active')

--Numero di brick (parametro linea da selezionare)

select count(*) from (select distinct brick_code from struttura_terr where line_number=#LINEA# and terr_status='Active')

--RSM non definiti

select count(*) from (select distinct am_no from struttura_terr where rsm_no ='UD' and line_number in (#LINEA#) and terr_status='Active')

--Tot. percentuali di condivisione (parametro linea da selezionare)

select sum(share_cond_brick) from struttura_terr where line_number=#LINEA# and terr_status='Active'

MEDICI VISITATI

--N medici non presenti in schedario

select count(*) from (select distinct med_id from visite where med_klist_id ='1' and vis_status='Active')

--N medici non riconosciuti

select count(*) from (select distinct isf_id from visite where med_id ='1' and vis_status='Active')

VENDITE AGLI OSPEDALI

--Numero ospedali non riconosciuti

select count(*) from (select distinct hosp_no from sales where hosp_id ='1' and sal_status='Active')

--Numero prodotti non riconosciuti

select count(*) from (select distinct mat_no from where mat_id ='1' and sal_status='Active')

--N ospedali non associati ad ISF

select count(*) from (select distinct hosp_no from sales where isf_no ='UD' and sal_status='Active')

NUMERO DI CAMPIONI

--N medici non riconosciuti

select count(*) from (select distinct isf_id from samples where samp_med_id ='1' and samp_status='Active')

--N saggi non presenti in anagrafica

select count(*) from samples where samp_id ='1' and samp_status='Active'

--N moduli con + di 4 saggi

select count(*) from (select count(*) tot, mod_id from samples where samp_status='Active' group by mod_id) where tot > 4

--N medici con + di 10 saggi

select count(*) from (select count(*) , samp_med_id from samples where samp_status ='Active' and samp_med_id <> 1 group by samp_med_id) where tot > 10

COSTI DI GESTIONE

--N profit center mancanti

select count(*) from (select distinct profit_center from load_Expenses minus select distinct profit_center from expenses_Alloc)

--N internal order mancanti

select count(*) from (select distinct internal_order from load_Expenses minus select distinct internal_order from expenses_Alloc)

--N prodotti mancanti

Select count(*) from (select distinct mat_no from material where flag_pl='Y' and mat_status= 'Active' minus select distinct mat_no from load_expenses)

--N di ISF mancanti

select count(*) from (select distinct isf_no from struttura_terr where terr_status='Active' minus select distinct isf_no from expenses_alloc)