

UNIVERSITÀ DEGLI STUDI DI PISA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
FACOLTÀ DI ECONOMIA

CORSO DI LAUREA SPECIALISTICA IN INFORMATICA
PER L'ECONOMIA E L'AZIENDA



Una soluzione di Business Intelligence per l'analisi del budget

Candidati

Marco Luchini

Matteo Pardini

Tutore aziendale

Dott. Stefano Celati

Relatore

Prof. Antonio Albano

ANNO ACCADEMICO 2010/11

Riassunto

La stesura di un adeguato piano annuale di budget, unito alla pianificazione e controllo di tutte le risorse aziendali interessate, sono divenute negli anni tema di importanza strategica e competitiva per tutte le aziende, a maggior ragione per quelle che operano in settori altamente evolutivi come quello tecnologico. L'attenzione si concentra quindi sull'allineamento delle attività e funzioni aziendali con gli obiettivi strategici d'impresa e sul continuo adeguamento ai sempre più frequenti mutamenti ambientali che la circondano.

Il progetto oggetto di questa tesi prevede la realizzazione di un'applicazione di budgeting e planning per un'azienda che opera nel settore informatico e tecnologico. La finalità è lo sviluppo, attraverso opportuni strumenti di analisi dei dati, di un sistema che permetta ad uno o più utenti incaricati di visualizzare, gestire ed analizzare le informazioni riferite ad un piano budgetario annuale. Per raggiungere questo scopo si renderà necessario definire uno o più data mart.

Indice

Riassunto	3
1 Introduzione	9
1.1 Rassegna della letteratura	10
1.2 Contenuto della tesi	10
2 Il budget	13
2.1 Lo strumento budget	13
2.2 Il budget commerciale	18
2.3 Determinazione del punto di pareggio	23
2.4 Linee guida alla progettazione di un data warehouse per il budget	27
2.4.1 Premessa	28
2.4.2 Schema del fatto budget	30
2.4.3 Schema del fatto impegni	32
2.4.4 Schema del fatto pagamenti	33
2.4.5 Osservazioni conclusive	34
2.5 La struttura di un report per il budget	35
2.6 Riassunto	37
3 Progettazione del data warehouse	39
3.1 Definizione del problema e analisi dei processi	39
3.2 Progettazione del data mart dei costi a budget	43
3.2.1 Raccolta dei requisiti di analisi	43
3.2.2 Specifica dei requisiti	44
3.2.3 Progettazione concettuale del data mart	46

3.3	Progettazione del data mart dei ricavi a budget	47
3.3.1	Raccolta dei requisiti di analisi	47
3.3.2	Specifica dei requisiti	47
3.3.3	Progettazione concettuale del data mart	49
3.4	Progettazione del data mart dei costi a consuntivo	50
3.4.1	Raccolta dei requisiti di analisi	50
3.4.2	Specifica dei requisiti	50
3.4.3	Progettazione concettuale del data mart	53
3.5	Progettazione del data mart dei ricavi a consuntivo	53
3.5.1	Raccolta dei requisiti di analisi	53
3.5.2	Specifica dei requisiti	54
3.5.3	Progettazione concettuale del data mart	56
3.6	Progettazione logica dei data mart	57
3.7	Osservazioni conclusive	61
3.8	La tabella per l'analisi del punto di pareggio	68
3.9	Riassunto	71
4	I flussi ETL	73
4.1	Lo strumento Pentaho Data Integration	73
4.2	Estrazione, trasformazione e caricamento dei dati	77
4.2.1	ETL: data warehouse	78
4.2.2	ETL: punto di pareggio	81
4.3	Riassunto	87
5	Le analisi dei dati	89
5.1	Il sistema di analisi dei dati	89
5.2	Il modello multidimensionale a cubo	93
5.2.1	Gli strumenti della suite Pentaho C.E.	93
	Mondrian	94
	Schema Workbench	95
	JPivot	98
	Saiku	101
5.3	Esempi di analisi dei dati	105
5.3.1	Analisi sui dati di costo	105
5.3.2	Analisi sui dati di ricavo	109

5.3.3	Analisi complessiva dei dati	109
5.4	Osservazioni conclusive	112
5.5	Riassunto	116
6	Il sistema di reporting	119
6.1	Gli strumenti della suite Pentaho C.E.	119
	Report Designer	120
	File Xaction	132
6.2	Rapporti basati sui requisiti di analisi	141
	Rapporti per l'analisi dei dati di costo	141
	Rapporti per l'analisi dei dati di ricavo	142
	Rapporti per l'analisi complessiva dei dati	143
6.3	Rapporti grafici per le analisi prestazionali	144
	Rapporti relativamente semplici	144
	Rapporti relativamente complessi	146
	Rapporti complessi senza l'SQL analitico	148
6.4	Rapporto per l'analisi del punto di pareggio	152
6.5	Osservazioni conclusive	155
6.6	Riassunto	157
7	Le tecnologie Open Source utilizzate	159
7.1	Risorse e relazioni del sistema	159
	7.1.1 La Suite Pentaho Community Edition	161
	7.1.2 SugarCRM, il Customer Relationship Management	166
	7.1.3 Lo strumento PortaBI	169
7.2	Riassunto	170
	Conclusioni	173
A	Ulteriori esempi di report e grafici	177
	Rapporti basati sui requisiti di analisi	177
	Grafici per analisi prestazionali	182
	Bibliografia	193
	Ringraziamenti	195

Capitolo 1

Introduzione

In questa tesi si presenta il progetto “Una soluzione di Business Intelligence per l’analisi del budget”, un servizio espressamente richiesto da un cliente dell’azienda presso cui abbiamo svolto la nostra attività. Il problema si presenta dal momento in cui il cliente decide di voler estendere il proprio sistema gestionale con un nuovo strumento per la pianificazione e budgeting che rispecchi il più possibile lo standard finora utilizzato dal personale amministrativo. L’importanza della richiesta risiede non tanto nella possibilità di trattare le informazioni sotto un nuovo punto di vista, quello budgetario, ma soprattutto nella potenzialità di sfruttare tali informazioni confrontandole con il consuntivo consolidato, derivandone così scostamenti ed analisi esplicative della corrente gestione aziendale. Per realizzare quanto richiesto si è pensato di sviluppare il progetto per fasi successive. Partendo quindi dallo studio delle tematiche relative alla definizione di piani di budget commerciali annuali per le imprese, si è passati alla realizzazione di una base di dati appositamente dedicata, in grado di mantenere le informazioni necessarie a riguardo. Questi dati costituiranno il deposito centrale delle informazioni di cui l’utente avrà bisogno per definire le analisi multidimensionali successive. Tali analisi sono rese possibili grazie alla progettazione di *cubi OLAP* che ricevono in input i dati provenienti dal *data mart*, la cui realizzazione è affidata alla seconda fase di progettazione. L’ultima fase riguarderà invece la creazione di reportistica ad hoc, che verrà utilizzata dalla direzione aziendale, ai fini della comunicazione interna.

La tesi è stata svolta presso l’azienda *Bnova S.r.L.* una società che si

occupa da anni di consulenza e system integration, focalizzando le proprie conoscenze nelle aree di sviluppo e progettazione *Data Warehouse* (dwh), *Business Intelligence* (BI), *Enterprise Content Management* (ECM) e *Customer Relationship Management* (CRM) utilizzando principalmente risorse *open source*. A tale scopo l'azienda ha messo a nostra disposizione la strumentazione software adeguata al raggiungimento dell'obiettivo, fornendoci una metodologia di lavoro, materiali e documenti informativi sull'argomento, oltre alle conoscenze e all'esperienza di tutta la squadra di consulenti, analisti e tecnici coinvolti.

1.1 Rassegna della letteratura

Per quanto riguarda le tematiche inerenti i sistemi informativi aziendali, il controllo di gestione nelle imprese e la pianificazione del processo di budgeting aziendale i testi consultati sono [Marchi 08] e [Paolini 04]. Altri testi interessanti per approfondimenti sono [Castellano 03] per il controllo di gestione e [Cinquini 08] per l'analisi dei costi. I sistemi informatici operativi e direzionali, la Business Intelligence, la reportistica e la progettazione di data warehouse sono discussi nel testo [Albano 09a], mentre i fondamenti di basi di dati sono trattati in [Albano 05b]. Il testo di riferimento con le linee guida per lo sviluppo di dwh inerenti il budget sono reperibili su [Adamson 98]. Aspetti interessanti riguardanti le potenziali analisi sviluppabili sul budget e sulla gestione della forza di vendita sono consultabili su [Barocco 09] e [Dalli 07]. Per l'utilizzo dello strumento di BI (Pentaho Community Edition) e di tutti i moduli utili di cui esso è composto ci siamo avvalsi delle guide online rilasciate dalla community di sviluppatori [CWH 11] e del manuale [Bouman 09].

1.2 Contenuto della tesi

Nel dettaglio la tesi si articola in Capitoli e Sezioni di seguito presentati:

- Il Capitolo 2 affronta le tematiche di pianificazione e sviluppo di un processo di budgeting, focalizzando l'attenzione sui budget operativi

dell'area commerciale, sulle loro possibili rappresentazioni, sulle informazioni e sui dati utili che questi devono mantenere. Si mostra un classico esempio, ampiamente discusso in letteratura, che riporta una possibile soluzione nella progettazione dei data mart utili alla gestione dei dati di budget. I successivi Capitoli trattano dettagliatamente tutte le fasi del progetto nell'ordine in cui queste sono state affrontate.

- Il Capitolo 3 presenta il caso di studio con le specifiche desiderate dall'azienda cliente e l'obiettivo che si vuole raggiungere. Si affronteranno quindi tutte le fasi di progettazione dei data mart a partire dall'analisi dei processi, dalla raccolta e dalle specifiche dei requisiti di analisi, fino ad arrivare allo sviluppo dello schema concettuale e logico.
- Il Capitolo 4 tratta nel dettaglio la creazione dei flussi ETL (*Extrat, Transform and Load*) necessari al popolamento del data warehouse progettato nella fase precedente. Si discuterà ampiamente la distinzione tra i flussi che caricano i dati nel data warehouse e quelli che invece a partire da quest'ultimo popolano la tabella per l'analisi del punto di pareggio. Si mostrano alcuni esempi di utilizzo degli strumenti adeguati forniti dalla piattaforma Pentaho C.E.
- Il Capitolo 5 è incentrato sull'attività di progettazione dei cubi OLAP per l'analisi dei dati. Si mostrano nel dettaglio gli strumenti utilizzati allo scopo ed i risultati di output ottenuti facendo uso di alcuni esempi.
- Il Capitolo 6 raccoglie e presenta gli output del passo precedente sotto forma di documenti di *report*, *dashboard* e grafici dinamici (basati su librerie *Flash* e *JavaScript*) con i quali l'utente può interagire direttamente in rete tramite apposite interfacce. Per ciascun esempio mostrato si riporta la relativa interrogazione SQL che estrae i risultati dal data warehouse, distinguendole in tre categorie principali: interrogazioni relativamente semplici, interrogazioni relativamente complesse e interrogazioni complesse senza l'SQL analitico. Si presentano inoltre le caratteristiche degli strumenti open source utilizzati allo scopo e le varie fasi di costruzione di un rapporto tramite un classico esempio.
- Il Capitolo 7 è dedicato alla presentazione delle risorse e tecnologie open source utilizzate per lo sviluppo del progetto, distinguendo la Sezione inerente alla suite Pentaho C.E., lo strumento di BI, da quella riferita

al prodotto *PortaBI* che integra le funzionalità di *web analysis*, *web reporting*, *web dashboards* e *data mining* utilizzato dal cliente finale. Si riporta anche una Sezione dedicata al prodotto *SugarCRM*, il CRM operativo di riferimento per *PortaBI*.

La tesi termina con le conclusioni finali sull'intero lavoro svolto, su come questo possa essere migliorato ed esteso, e con alcune considerazioni sull'esperienza maturata presso l'azienda durante l'attività collaborativa svolta.

Capitolo 2

Il budget

Chi non si è mai imbattuto in questa parola, soprattutto visti i tempi che corrono? Certamente chi frequenta ambienti economico-aziendali si troverà continuamente a confrontarsi con questo termine, ma anche chi non è del settore non avrà avuto difficoltà ad incappare, attraverso una qualsiasi fonte di informazione, in questo vocabolo. Purtroppo cercare di darne una definizione precisa ed univoca risulta essere un'impresa ardua oltre che riduttiva, considerato anche che, a seconda del contesto in cui è inserito, può assumere svariate sfaccettature.

Perciò in questo Capitolo si procederà con introdurre la tematica del budget comprendendone l'utilità e l'efficacia di una sua redazione all'interno dell'azienda. Solo successivamente si presenterà la tipologia che ha riguardato il nostro lavoro: *il budget commerciale*.

Per iniziare basterà sapere che con il termine budget si intende uno strumento interno attuo alla programmazione e al controllo delle risorse aziendali.

2.1 Lo strumento budget

Al giorno d'oggi il processo budgetario rappresenta lo strumento in possesso dei dirigenti per la pianificazione, il controllo e la verifica delle attività aziendali. È infatti attraverso questo procedimento, di analisi anticipata delle entrate e delle uscite future, che l'amministratore spera di conseguire un determinato livello di profitto guidando le operazioni delle unità aziendali.

La produzione di un piano di budget nasce dall'esigenza di *pianificazione*: cioè dalla necessità di monitorare e verificare l'attività dell'azienda, perché attraverso una corretta pianificazione si riducono i rischi e le incertezze future e soprattutto si consente all'impresa di affrontare nel modo ottimale eventuali imprevisti. Per renderla quanto più efficace, deve appoggiarsi su di un sistema informativo atto a rilevare ogni operazione gestionale, e a consentire di individuare possibili divergenze tra l'andamento effettivo e quello stabilito nel piano gestionale. Le funzionalità che il budget esplica, rappresentate in Figura 2.1 [Ciambotti 10], si distinguono in:

- *Controllo*: durante il processo budgetario è in atto una forma di controllo basata sul raffronto dei dati previsti con i corrispettivi dati consuntivi; gli eventuali scostamenti, positivi o negativi, possono comportare azioni correttive. Ne consegue quindi che l'azienda deve essere in grado di misurare i propri risultati operativi in tempi brevi.
- *Verifica*: una volta terminato il processo, i dati consolidati rappresentano il riferimento per i responsabili incaricati di prendere le decisioni, e la base da cui partire per future redazioni.

All'interno dell'azienda è la figura del top management che dà inizio al processo budgetario, richiedendo all'organizzazione la preparazione di un piano per le operazioni future; ogni unità operativa (amministrativa, marketing-commerciale, produzione, ricerca ecc.) sviluppa il proprio piano operativo secondo gli obiettivi fissati dal top management. In seguito i singoli dipartimenti sviluppano un piano dettagliato delle azioni da intraprendere per raggiungere tali scopi.

Formalmente si presenta sotto forma di un documento (cartaceo o elettronico) in cui a fronte di un'analisi costi-ricavi i responsabili definiscono gli obiettivi che ciascuna area aziendale deve raggiungere nel periodo pre-stabilito. Generalmente si fa riferimento all'esercizio futuro ma può variare soprattutto in funzione del settore target dell'azienda. A tal fine basti pensare a quelle realtà i cui prodotti commercializzati risentono della stagionalità; in queste circostanze la definizione del budget avviene su intervalli ben specifici pari alla durata di ogni finestra temporale di vendita. Infine una volta terminata la stesura del budget, anche dopo aver subito una o più revisioni,

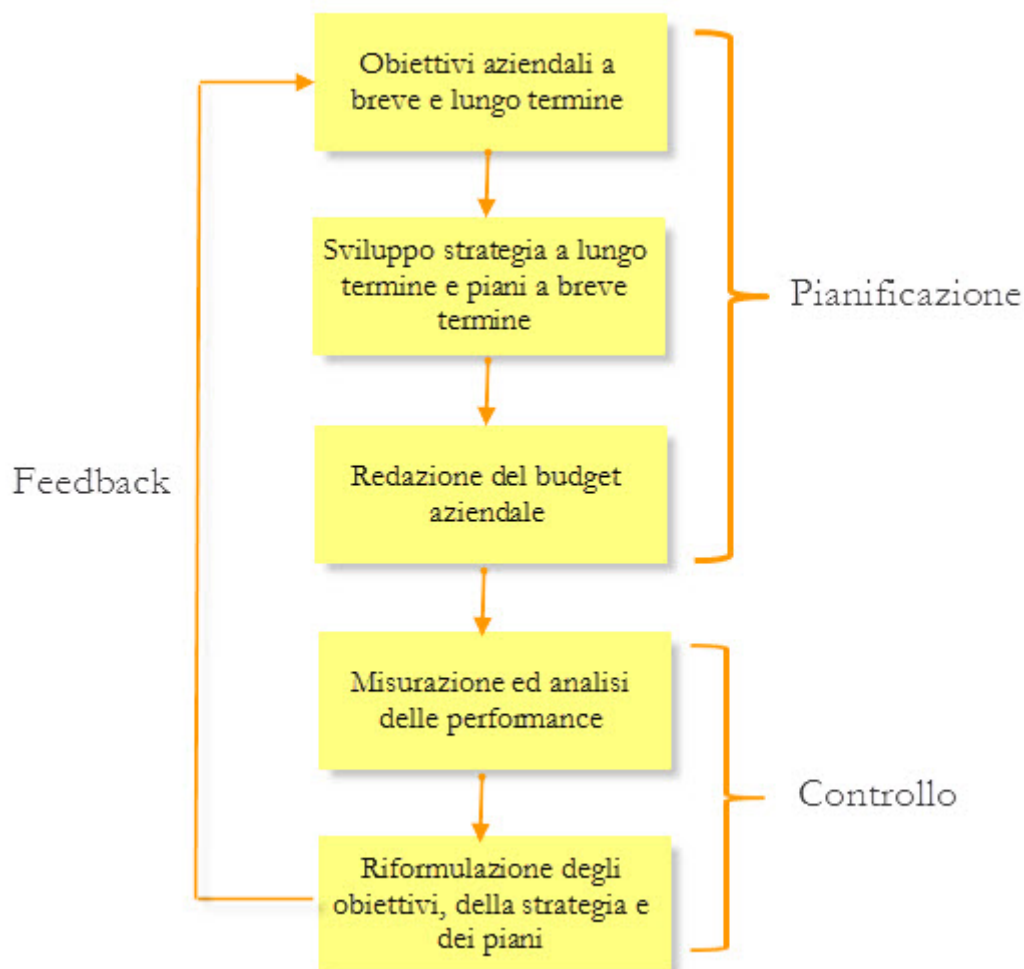


Figura 2.1: Fasi del budget

questo viene firmato dal redattore e passato per l'approvazione finale ai dirigenti del livello superiore, i quali possono prima di certificarlo definitivamente apportare modifiche o reinstrararlo nuovamente ad una nuova elaborazione al livello sottostante. Quando finalmente viene approvato, il suo contenuto rappresenta le linee guida per il periodo temporale prestabilito. Durante l'arco temporale interessato, il budget è sottoposto ad un continuo controllo incrociato con i rispettivi valori a consuntivo, e se necessario revisionato in corso d'opera.

Le funzionalità che questo strumento assolve non si limitano solamente a quelle della gestione economica-finanziaria (previsione costi-ricavi, analisi scostamenti, verifiche e correttivi in corso d'opera), anche se risultano essere

le principali, ma coinvolgono anche l'aspetto organizzativo della struttura e del personale aziendale. Tra queste si può individuare una predisposizione a migliorare l'integrità e l'efficienza aziendale, e più precisamente:

- Far riflettere sulle attività di propria competenza e quindi a responsabilizzare e ad impostare programmi specifici e monitorati continuamente.
- Promuovere la comunicazione tra le diverse *A.S.A.* (aree strategiche d'affari) portando al confronto su fabbisogni spesso contrastanti.
- In conseguenza al punto precedente a fornire dei criteri di valutazione delle performance condivisi.

A seconda della specifica area aziendale di riferimento o del tipo di analisi da svolgere, nel panorama attuale esistono molteplici tipologie di budget, principalmente suddivisi in due macro-categorie: *settoriali* e di *sintesi*. I primi riguardano i documenti relativi alle singole aree funzionali dell'azienda; i secondi si generano a partire dai budget settoriali e riguardano l'azienda a livello globale, e generalmente rappresentano una sintesi del bilancio a preventivo.

Nell'insieme dei budget settoriali si distinguono le seguenti tipologie:

- *Approvvigionamenti*: predisposto dall'ufficio acquisti riguarda la quantità ed il valore delle materie prime ed accessorie che sono coinvolte nel budget produttivo. Avviene partendo dalla somma degli articoli presenti nelle varie distinte base dei prodotti oggetto del budget produttivo e valorizzati al costo standard.
- *Investimenti*: raccoglie le esigenze che scaturiscono sia dal budget commerciale che da quello di produzione. Nasce ad esempio dalla necessità di voler entrare in una nicchia di mercato non ancora coperta; oppure dall'acquisizione di una nuova linea di montaggio per la produzione di un articolo innovativo e non ancora commercializzato. Diventa fondamentale la redazione di un budget di questo tipo attuato ad un investimento mirato, ponderando la politica industriale e commerciale dell'azienda.
- *Struttura*: rappresenta il budget di tutto il personale: produttivo, commerciale, amministrativo, di ricerca, dei servizi comuni ed informativi e viene espresso sia in termini di numero addetti per centro di responsabilità che per costo annuo.

- *Area di produzione*: serve a quantificare i programmi di produzione e ha una duplice funzione:
 - *Reddituale*: in quanto determina la categoria dei costi di produzione derivanti dal processo di acquisizione e consumo dei fattori produttivi necessari alla realizzazione del prodotto da commercializzare (materie prime, energia, materiali di consumo, mano d'opera, spese di manutenzione, ecc.).
 - *Investimento*: la parte derivante dagli investimenti in immobilizzazioni materiali (immobili, macchinari e impianti) e immateriali (studi di ricerche e sviluppo, brevetti industriali, spese di impianto).
- *Vendite o commerciale*: si riferisce al programma delle vendite e dei relativi ricavi, nonché dei costi commerciali. Sostanzialmente è un piano degli ordini da acquisire nel periodo preso in osservazione. Anche in questa circostanza possiamo individuare una duplice dimensione di analisi:
 - *Reddituale*: fornisce i margini che derivano dalla sintesi dei ricavi lordi che si prevede di ottenere dalla vendita dei prodotti/servizi, suddivisi per linee, canali ed aree di vendita, cliente, ecc. e dei costi commerciali sostenuti per la commercializzazione dei prodotti/servizi offerti.
 - *Investimento*: determina gli investimenti da operare nel settore commerciale, necessari per portare a compimento i ricavi lordi stabiliti.

Passando invece alla categoria dei budget di sintesi si possono distinguere le seguenti tipologie:

- *Finanziario*: determina la coerenza sostanziale e la fattibilità finanziaria dell'insieme dei programmi d'azione settoriali. Il budget finanziario viene suddiviso in due prospetti:
 - *Fonti e impieghi*: in cui si evidenziano le fonti e gli impieghi finanziari nella gestione reddituale, nella gestione delle attività di investimento e nella gestione delle attività di finanziamento.

- *Tesoreria*: esprime la compatibilità di una certa politica di investimento e di finanziamento sotto il profilo della coerenza fra i tempi di erogazione delle entrate rispetto a quelli derivanti dalla necessità delle uscite.
- *Stato patrimoniale*: è uno stato patrimoniale previsionale che segnala le strutture delle attività e passività in essere al termine del periodo considerato.
- *Economico*: sintetizza in un unico prospetto l'insieme dei ricavi e dei costi evidenziando il risultato operativo della gestione tipica dell'azienda. A ciò si aggiungono, una volta completato il budget finanziario, gli oneri ed i proventi finanziari, elementi di gestione straordinaria (se prevedibili) e le imposte, al fine di determinare un risultato netto. La struttura mediante cui viene espresso il risultato economico consente di evidenziare anche una serie di risultati intermedi come i margini di contribuzione, il risultato caratteristico e il risultato ante imposte (solitamente espressi con le sigle MC, EBIT e EBITDA).

Concludendo è importante sapere che “il budget deve essere redatto con *chiarezza* e deve rappresentare in modo *veritiero* e *corretto* gli obiettivi che si vogliono raggiungere”. Questo documento deve cioè dare una rappresentazione reale di tutte le possibili variabili del sistema azienda, facilitandone la comprensione agli utenti cui è rivolto e presentando obiettivi realistici e raggiungibili.

Ritornando al progetto e al caso di studio di riferimento, si dedica la successiva Sezione all'analisi del budget commerciale, oggetto della richiesta da parte dell'azienda cliente.

2.2 Il budget commerciale

Il budget commerciale riguarda quelle imprese o parti di aziende che non producono in proprio beni a partire da materie prime o sussidiarie, ma bensì che trattano la vendita di merci o servizi direttamente al cliente (si pensi ad un grossista il cui compito è solo quello di distribuire un prodotto sul mercato, o ad un distributore di servizi/consulenze o infine al reparto commerciale di una qualsiasi azienda). Pertanto queste realtà, in assenza del processo

produttivo, presentano una struttura molto più snella e quindi anche la redazione del budget risulta semplificata.

Il budget commerciale esprime quindi gli obiettivi e i programmi dell'area commerciale in termini di volumi di vendita e conseguenti ricavi, considerando i costi legati alla commercializzazione dei prodotti e/o servizi. Per poter passare dal programma delle quantità di vendita al budget dei ricavi occorre valorizzare le quantità fisiche ad un determinato prezzo unitario; questo rappresenta per la direzione aziendale uno dei problemi più complessi a causa delle numerose variabili coinvolte. In Figura 2.2 è riportato un classico esempio di composizione del budget commerciale.

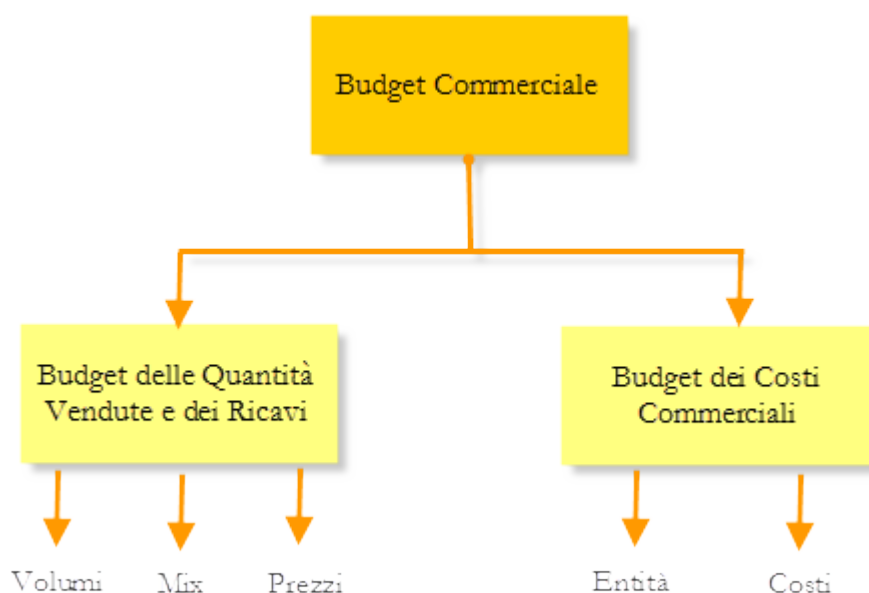


Figura 2.2: Composizione del budget commerciale

Possiamo quindi definire il budget dell'area commerciale come una composizione tra il budget delle vendite e i costi legati all'attività commerciale: il primo evidenzia l'ammontare quantitativo delle unità che si prevede di vendere nel prossimo periodo (*mix*) fissando un prezzo di vendita unitario per ciascun tipo di prodotto (*prezzo*) così da prevedere i ricavi conseguiti; il secondo individua le varie tipologie di costi sostenuti dall'azienda per la commercializzazione dei prodotti, determinando il quantitativo dei suddetti costi.

La definizione delle quantità che si prevede di vendere vengono determinate con varie metodologie [Dalli 07]:

- **Metodi soggettivi**

Sono legati all'intuizione e all'esperienza dei dirigenti, degli esperti esterni e dei venditori. Si distinguono in 4 tipologie:

- *Opinione dei dirigenti*: è il più tradizionale e semplice, si basa sulle valutazioni dei top manager e molto spesso sono basate sull'esperienza, l'osservazione e l'intuito. Non avendo una base scientifica risulta una tecnica semplice e veloce, ma implica un notevole dispendio di energie da parte dei dirigenti coinvolti ed inoltre comporta il rischio che i manager più influenti dominino le decisioni.
- *Metodo Delphi*: viene selezionato un panel esterno di esperti del settore, a cui si chiede la formulazione di una previsione di vendita. Tutte le stime raccolte dal responsabile del procedimento vengono riassunte attraverso una media e riproposte al panel, al quale viene chiesto di formulare una nuova previsione. Questo processo si ripete fino a quando non si giunge ad una conclusione comune. Si riesce così a minimizzare gli effetti dovuti alle dinamiche di gruppo.
- *Contributo della forza vendita*: ad ogni venditore viene chiesto di stimare la propria previsione di vendita; in seguito vengono sommate tutte assieme per ottenere il totale previsto a livello aziendale. È un buon metodo, in quanto si presume che gli agenti conoscano le potenzialità dei loro clienti, ma purtroppo spesso i venditori per interessi personali tendono a sottostimare le aspettative di vendita.
- *Intenzioni d'acquisto*: si contattano alcuni potenziali clienti, interrogandoli riguardo alle loro intenzioni d'acquisto future e trasladole sulla previsione di vendita. Basandosi su informazioni ottenute direttamente dalle persone interessate risulta un metodo con una buona affidabilità, purtroppo però ha un processo lungo e spesso economicamente dispendioso.

- **Metodi oggettivi**

Tendono ad eliminare la componente soggettiva, basandosi su analisi quantitative. In questa tipologia si distinguono:

- *Test di mercato*: molto utilizzati nei casi in cui si effettui il lancio di un nuovo prodotto. Consistono nel piazzare il prodotto su diverse aree geografiche rappresentative e nel proiettare poi i risultati ottenuti sull'intero mercato. Sono parecchio costosi e richiedono quindi un notevole sforzo economico e organizzativo.

- *Media mobile*: le previsioni di vendita vengono calcolate a partire dalla media delle vendite effettuate negli anni precedenti. Questo approccio risulta semplice, ma bisogna prestare attenzione alla determinare degli appropriati periodi di riferimento; inoltre assegna lo stesso peso ad ogni periodo preso in considerazione.

$$Vendite\ previste_{t+1} = (vendite_t + vendite_{t-1} \dots vendite_{t-n+1})/n$$

- *Riduzione esponenziale*: è una specie di media mobile ma con la possibilità di attribuire un peso differente alle vendite generate nei vari anni, grazie alla costante di riduzione α . Alti valori di α assegnano un peso maggiore alle vendite più recenti.

$$Vendite\ previste_{t+1} = \alpha vendite_t + (1-\alpha) vendite_{t-1}$$

- *Scomposizione*: attraverso questa procedura si tende ad isolare le quattro componenti presenti all'interno della serie storica di vendita: il trend, il ciclo, la stagionalità e il fattore aleatorio.

- *Regressione semplice*: si rappresenta attraverso una linea retta la relazione tra la variabile dipendente Y (ad esempio le vendite) e la variabile indipendente X (ad esempio il tempo). Si determinano i coefficienti a e b della retta che meglio si adattano ai dati, attraverso il metodo dei minimi quadrati: $Y = a + b * X$.

Una volta stabilite le quantità previste di ogni bene, si fissa il loro prezzo di vendita e si procede alla stesura del budget delle vendite. Può essere articolato in relazione a sottoperiodi rispetto a quello di riferimento, questo per venire incontro alle particolarità fortemente stagionali di alcuni mercati, per effettuare previsioni più accurate e per garantire il riesame della posizione

dell'impresa più volte, diminuendo così la probabilità che le varie operazioni sfuggano al controllo. L'articolazione del budget commerciale può essere ulteriormente approfondita aggregando le vendite in funzione di dimensioni personali ritenute significative per esigenze di carattere strategico, ad esempio le più comuni sono: per cliente, per prodotto o famiglia di prodotti, per canale distributivo, per agente di vendita, per area geografica, per A.S.A. e per distribuzione temporale [Marchi 08].

	GENNAIO		FEBBRAIO		MARZO		I TRIMESTRE		APRILE		TOTALE
PRODOTTO 1	Quantità	Prezzo	Quantità	Prezzo	Quantità	Prezzo	Quantità	Prezzo	Quantità	Prezzo		
Zona A												
Zona B												
Zona C												
PRODOTTO 2												
Zona A												
Zona B												
Zona C												
PRODOTTO 3												
Zona A												
Zona B												
Zona C												

ZONA A	VENDITORE X				VENDITORE Y				VENDITORE Z			
	I TRIM	II TRIM	III TRIM	IV TRIM	I TRIM	II TRIM	III TRIM	IV TRIM	I TRIM	II TRIM	III TRIM	IV TRIM
PRODOTTO 1												
PRODOTTO 2												
PRODOTTO 3												

Figura 2.3: Tipiche rappresentazioni del budget delle vendite

Per quanto riguarda il budget dei costi si procede ad individuare quelli attinenti l'attività commerciale, classificandoli in due macro categorie:

- *Costi fissi*: i cui ammontare sono indipendenti dall'output prodotto (ad esempio la componente fissa dello stipendio di un agente, i canoni di affitto di un immobile ecc.).
- *Costi variabili*: i cui ammontare dipendono direttamente dalle quantità di output prodotte (per esempio le provvigioni dei venditori, le spese telefoniche o delle utenze ecc.).

I costi fissi non essendo imputabili direttamente ai prodotti oggetto della vendita, vengono ripartiti secondo delle *basi di riparto* che ne attribuiscono una quota ad ogni bene in funzione di alcuni parametri: i due principali metodi di suddivisione sono in proporzione delle quantità previste o del fatturato di vendita.

Quando sono definiti a budget anche i costi, si procede alla stesura del piano del budget commerciale, come mostrato nell'esempio di Figura 2.4.

	PRODOTTO 1	PRODOTTO 2	TOTALE
BUDGET DELLE VENDITE			
Quantità programmate	1.600.000	1.700.000	3.300.000
Prezzo unitario	€ 28,00	€ 25,00	
RICAVI DI VENDITA	€ 44.800.000,00	€ 42.500.000,00	€ 87.300.000,00
BUDGET COSTI VARIABILI			
Provvigione per pezzo	0,02%	0,01%	
Provvigioni totali	€ 8.960,00	€ 4.250,00	€ 13.210,00
Imballaggio per pezzo	€ 0,05	€ 0,10	
Imballaggi totali	€ 80.000,00	€ 170.000,00	€ 250.000,00
TOTALE COSTI VARIABILI	€ 88.960,00	€ 174.250,00	€ 263.210,00
BUDGET COSTI FISSI (ripartiti sui prodotti 1 e 2 in base al loro fatturato)			
spese stipendi agenti annuo € 720.000,00	€ 369.484,54	€ 350.515,46	€ 720.000,00
spese promozione annuo € 20.800,00	€ 10.674,00	€ 10.126,00	€ 20.800,00
TOTALE COSTI FISSI	€ 380.158,53	€ 360.641,47	€ 740.800,00
UTILE (RICAVI - COSTI TOTALI)	€ 44.330.881,47	€ 41.965.108,53	€ 86.295.990,00

Figura 2.4: Esempio di budget commerciale

A questo punto risulta interessante effettuare un'analisi gestionale del tipo *CVR* (costi-volumi-risultati) per determinare il *break even point*: punto di pareggio tra i costi sostenuti e i ricavi dell'azienda.

2.3 Determinazione del punto di pareggio

L'analisi *CVR* è un tipico modello di analisi del comportamento dei costi aziendali grazie al quale è possibile determinare il punto di pareggio fra costi e ricavi, generalmente in funzione della quantità di beni prodotti e venduti.

La quantità venduta, rappresentata tramite la variabile Q , definisce il *driver* di analisi. Posta la proporzionalità dei costi variabili con il *cost driver* Q , si evidenzia che il break even point (detto anche punto di rottura o di pareggio) ha luogo nell'istante in cui l'utile raggiunge un livello pari a 0.

Espresso in termini matematici, posto che deve valere:

$$UTILE = 0$$

e riscrivendo l'uguaglianza in termini di costi e ricavi si deve verificare:

$$UTILE = RT - CT = 0 \Rightarrow RT = CT$$

i cui singoli fattori definiscono i seguenti valori:

$$\left\{ \begin{array}{l} RT = \text{ammontare ricavi totali} \\ CT = \text{ammontare costi totali} \\ CF = \text{ammontare costi fissi} \\ CV = \text{ammontare costi variabili} \end{array} \right.$$

i membri possono essere così scomposti:

$$\begin{aligned} RT &= p * Q \text{ (quantità venduta al prezzo } p) \\ CT &= CF + CV \text{ con } CV = v * Q \text{ (quantità acquistata al costo } v) \end{aligned}$$

$$\left\{ \begin{array}{l} v = \text{costo variabile unitario} \\ p = \text{prezzo di vendita unitario} \end{array} \right.$$

sostituendo nell'espressione precedente si ottiene:

$$p * Q = CF + (v * Q)$$

ed infine risolvendo in funzione di Q si definisce la nuova uguaglianza:

$$Q^* = \frac{CF}{(p - v)}$$

$$\left\{ (p - v) = \text{margine di contribuzione unitario} \right.$$

Calcolato il valore Q^* si può procedere rappresentando su un grafico cartesiano le rette RT e CT, come riportato nella Figura 2.5, visualizzando così l'area di perdita ed eventualmente di utile. Il modello di analisi in oggetto si basa su alcune assunzioni importanti:

- I costi variabili sono direttamente proporzionali al driver delle quantità e pertanto possono essere rappresentati come una linea retta sul piano ($Y = a + b * X$) il cui coefficiente angolare è pari al costo variabile unitario.
- Il volume di produzione coincide con il volume di vendita (in altri termini si assume che l'intera produzione venga venduta), pertanto non si fa alcun riferimento alla gestione delle scorte del magazzino.
- Si ipotizza la linearità delle funzioni di costo e ricavo, il che comporta un andamento costante nel tempo dei prezzi, dei costi dei fattori di produzione e dei costi fissi.
- Per scopi rappresentativi è importante evidenziare una netta distinzione tra costi fissi e variabili anche se non sempre nella realtà è facile distinguerli rispetto al volume produttivo.

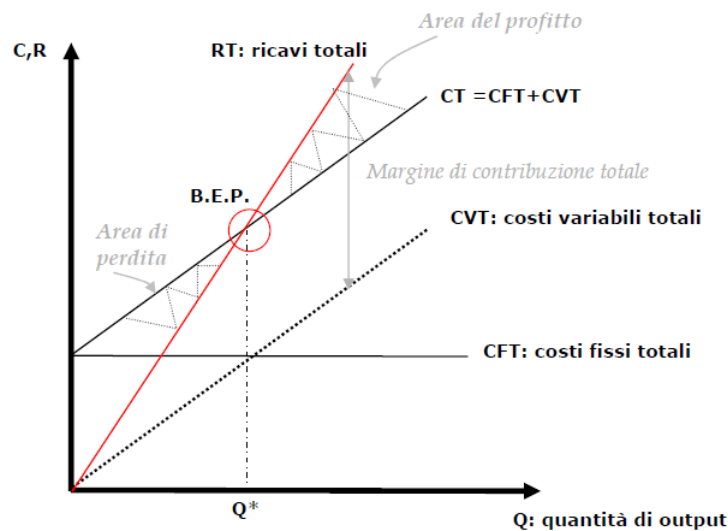


Figura 2.5: Analisi C-V-R con driver la quantità venduta

Dalla formula calcolata in precedenza ($Q^* = \frac{CF}{(p - v)}$), si osserva come il punto di pareggio varia proporzionalmente al variare dei costi fissi (all'aumentare di questi la quantità di equilibrio Q^* aumenta spostandosi verso destra sull'asse delle ascisse); al contrario varierà in modo inversamente proporzionale rispetto al margine di contribuzione (all'aumentare del margine la quantità di equilibrio Q^* diminuisce spostandosi verso sinistra sull'asse delle ascisse).

Per il caso di studio in oggetto, questo modello di analisi del break even point non trova applicazione, in quanto l'azienda cliente non vende direttamente dei prodotti ma piuttosto fornisce servizi e consulenze difficilmente quantificabili in termini di quantità prodotte Q . Si è quindi pensato di apportare alcune modifiche al modello appena descritto. Lasciando inalterati i livelli di costi e ricavi sull'asse delle ordinate, si va a modificare la quantità di produzione sull'asse delle ascisse sostituendola con l'intervallo temporale di riferimento (tipicamente l'anno solare espresso in giorni). Il nuovo grafico cartesiano per l'analisi CVR ottenuto può così essere espresso nella forma in Figura 2.6.

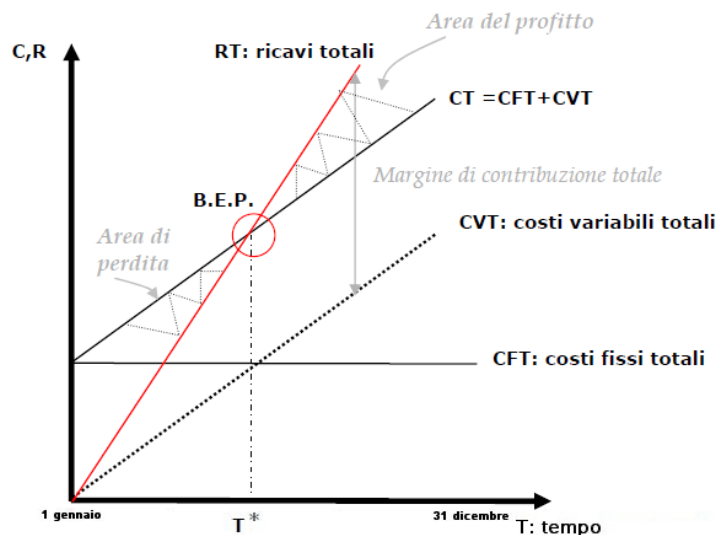


Figura 2.6: Analisi C-V-R con driver il tempo

L'obiettivo finale nell'utilizzo di questo nuovo modello di analisi è quello di poter stabilire il giorno dell'anno in cui l'azienda raggiungerà il suo punto di pareggio tra costi totali e ricavi totali (quindi un utile nullo).

All'inizio dell'anno corrente (1 gennaio) i dati di analisi saranno riferiti ai soli valori di budget, pertanto il punto di pareggio è da considerarsi esclusivamente una previsione fatta dal top management aziendale; una proiezione piuttosto che un dato certo e affidabile. Col trascorrere del tempo il sistema si popolerà di dati a consuntivo che prenderanno il posto dei corrispondenti valori di proiezione a budget; in questo modo l'analisi potrà essere rieseguita ogni volta sfruttando la porzione di dati consuntivati e la rimanente porzione

di dati di budget nella quantità utile al raggiungimento della data di fine anno (al 31 dicembre).

L'unione tra informazioni a consuntivo ed a budget è eseguita sfruttando un algoritmo previsionale di *forecast* definito con la collaborazione del committente. Più precisamente, l'algoritmo va a ridefinire i coefficienti angolari delle rette di costi e ricavi ponderando i valori a consuntivo e quelli a budget in funzione del numero di giorni. A tale scopo sfrutta la seguente formula:

$$Forecast = \frac{(X * \alpha) + (Y * \beta)}{\alpha + \beta}$$

le cui variabili indicate sono le seguenti:

$$\left\{ \begin{array}{l} X = \text{ammontare del costo/ricavo unitario giornaliero consuntivo} \\ \alpha = \text{massimo giorno del costo/ricavo rilevato a consuntivo} \\ Y = \text{ammontare del costo/ricavo unitario giornaliero budgetario} \\ \beta = \text{totale giorni di validità dei dati di budget} \end{array} \right.$$

Col trascorrere del tempo la porzione dei dati a consuntivo sarà sempre più numerosa, e si assegnerà un peso sempre maggiore a quest'ultimi rispetto a quelli a budget. Da quanto detto ne deriva che l'analisi tenderà sempre di più a rappresentare una situazione reale e sempre meno una situazione previsionale; fino al 31 dicembre quando tutti i dati rappresentati sul piano cartesiano saranno esclusivamente informazioni a consuntivo.

2.4 Linee guida alla progettazione di un data warehouse per il budget

Questa Sezione vuol essere un anello di congiunzione tra quanto finora esposto nel presente Capitolo, di natura prettamente teorico-economica, e quanto invece verrà presentato nel seguito, ovvero la progettazione del data warehouse per la gestione e l'analisi dei dati a budget e a consuntivo dell'azienda cliente. Pertanto nelle successive Sottosezioni andremo ad analizzare come la letteratura tratta le tematiche riguardanti la progettazione di strutture dati per la gestione del budget [Adamson 98].

2.4.1 Premessa

Come già ampiamente trattato in precedenza, si evidenzia come le imprese, senza distinzione di tipologia, dimensione o settore di appartenenza, tendano a programmare i loro piani di spesa e ricavi annuali con la creazione di un budget. Queste due componenti, costi e ricavi, spesso vengono considerati come concetti distinti sebbene abbiano una profonda influenza l'una con l'altra. Questo accade sia nelle grandi aziende dove lo sviluppo del piano budgetario può coinvolgere molti cicli di proposte e revisioni prima dell'approvazione finale, sia nelle piccole imprese dove molto spesso il budget può essere formulato abbastanza rapidamente. Dunque si rende necessario un inizio del processo stesso con congruo anticipo rispetto all'anno fiscale venturo di riferimento.

Vista la varietà degli scenari aziendali ogni budget viene redatto adattandolo alla propria realtà, distinguendosi quindi da quello delle altre imprese anche appartenenti allo stesso panorama. Nonostante ciò in tutti i piani budgetari è possibile riscontrare nella loro strutturazione un grado di somiglianza, esistendo delle linee comuni che prescindono dal tipo di impresa e dal settore d'appartenenza.

In questa Sezione ci focalizzeremo su questa struttura base per la progettazione del data warehouse del *budget riguardante i costi*, in quanto le stesse tecniche possono essere applicate anche alla parte dei ricavi. I principali obiettivi aziendali che il data warehouse del budget si prefigge di modellare sono:

- L'acquisizione dei valori delle voci del budget.
- L'acquisizione degli impegni intrapresi.
- L'acquisizione dei pagamenti fatti a fronte degli impegni.
- La tracciatura di eventuali revisioni che avvengono durante l'anno di riferimento.
- La possibilità di effettuare analisi comparative dei valori di: budget, impegni e pagamenti nel tempo, per linea di budget, dipartimento aziendale, conto economico movimentato.

Prima di cominciare la trattazione della fase di progettazione dobbiamo definire alcuni concetti importanti inerenti la tematica del budget, riportati nel seguito.

- *Le linee del budget*: è l'informazione minima presente nel budget. Ciascuna linea rappresenta l'autorizzazione per una determinata parte dell'azienda a spendere una somma di denaro per un particolare scopo durante un arco temporale. L'insieme delle linee di budget da origine al budget stesso. Ogni linea è caratterizzata da:
 - Le voci di bilancio che la spesa in oggetto andrà a movimentare (ad esempio le spese pubblicitarie). Quest'ultima risulta opzionale e spesso viene omessa in quanto ha rilevanza principalmente per i commercialisti.
 - Il dipartimento aziendale autorizzato a compiere questa spesa (ad esempio l'area marketing).
 - Il fine per cui è stata prevista questa spesa (ad esempio la produzione di volantini pubblicitari). Le aziende che effettuano budget molto dettagliati spesso dividono lo scopo in categoria e sottocategoria, per ottenere più livelli di aggregazione in fase di analisi.
 - L'intervallo di riferimento della spesa, che rappresenta la granularità temporale, di solito l'anno, il trimestre o il mese (difficilmente si scende a un livello inferiore, ma anche questo dipende dal tipo di attività aziendale che si intende monitorare).
 - L'importo espresso in valuta, che quantifica la somma prevista da spendere.
- *Gli impegni*: questo concetto serve per distinguere l'istante in cui nasce un'esigenza da soddisfare, spesso con l'acquisto di un bene o servizio, dal successivo momento in cui si ha l'effettiva manifestazione economica associata. Per fare un esempio chiarificatore supponiamo che il direttore di un ufficio ordini al fornitore un quantitativo di carta per la propria stampante. Al momento che l'ordine viene piazzato l'uscita monetaria di cassa non si è ancora verificata, infatti questa avverrà solo successivamente alla ricezione dell'oggetto, quando il venditore emetterà la fattura e conseguentemente l'ufficio ragioneria dell'azienda provvederà a dare conferma per eseguire il pagamento. Esiste quindi uno sfasamento temporale che, a seconda degli accordi tra il cliente ed il fornitore

sulle condizioni di pagamento, può essere anche di alcuni mesi. Risulta però evidente che nell'attimo stesso in cui l'ordine viene processato, anche se questo ancora non è stato spedito nè pagato, rivesta una notevole importanza ai fini del budget; infatti l'importo verso il venditore è da considerarsi oramai *impegnato* e deve essere sottratto dal valore residuo del budget. Letteralmente può essere tradotto con l'espressione "*l'impegno di*" e rappresenta un accordo formale tra due parti che si impegnano a scambiare un bene o servizio (il fornitore) in cambio di una contropartita monetaria (il cliente). Da qui scaturisce la definizione di impegno.

- *I pagamenti*: questo termine si riferisce alla manifestazione economica poc'anzi menzionata. Rappresenta dunque una transazione nella quale fondi monetari vengono trasferiti verso il fornitore in seguito alla ricezione di un bene o servizio. Quest'entità ricopre particolare importanza per la parte finanziaria e amministrativa di un'azienda.

Esiste quindi una corrispondenza temporale tra questi tre concetti appena indicati: inizialmente si stila il piano di budget, in seguito i reparti aziendali prendono degli impegni di spesa ed infine si procede al pagamento dell'impegno.



Figura 2.7: Conseguenzialità del processo di monitoraggio dei costi

Al termine di questa premessa risulta evidente come il processo di progettazione del data warehouse necessita la modellazione di tre processi: budget, impegni e pagamenti.

2.4.2 Schema del fatto budget

La granularità del fatto è rappresentata dall'importo previsto di spesa che un dipartimento può sostenere per la fornitura di un bene o servizio durante un mese dell'anno di riferimento e del relativo conto economico che il costo movimentata.

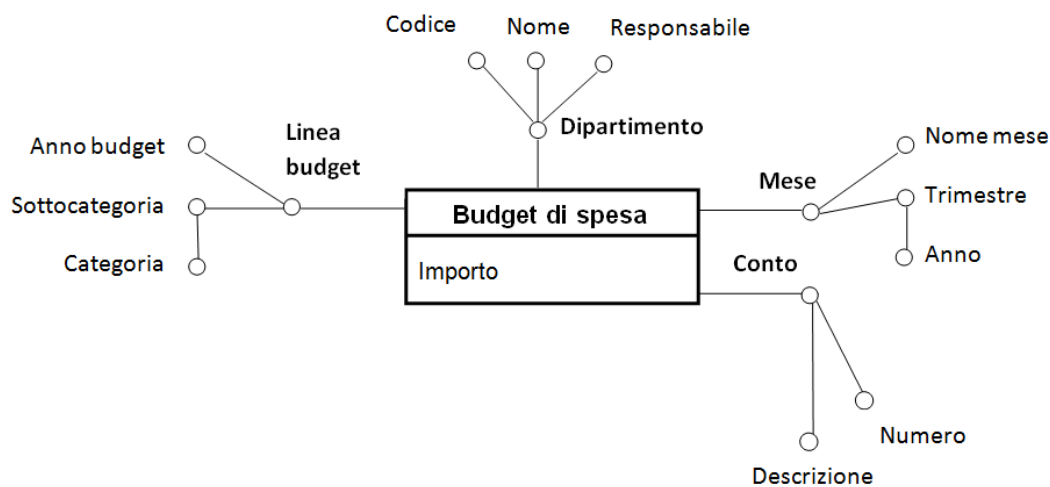


Figura 2.8: Schema concettuale del data mart del budget

Lo schema concettuale del data mart relativo al budget è il seguente di Figura 2.8. Durante la modellazione bisogna tenere in considerazione che:

- Nel corso dell'anno il budget può subire delle revisioni. Pertanto queste vengono implementate, all'interno della tabella dei fatti, andando ad aggiungere righe che si differenziano dalle precedenti soltanto per l'importo correttivo e per il mese in cui è avvenuta la revisione. In Figura 2.9 si riporta un esempio in cui inizialmente a novembre 2008 viene previsto un importo a budget di €200.000. Successivamente a luglio 2009 questo valore viene rivisto a €250.000, per far ciò si inserisce una nuova riga con importo pari a €50.000.

Conto	Dipartimento	Linea di budget	Anno budget	Mese	Importo
98-12-1092	Ufficio amministrazione	Fornitura per ufficio	2009	novembre-08	€ 200.000,00
98-12-1092	Ufficio amministrazione	Fornitura per ufficio	2009	luglio-09	€ 50.000,00

Figura 2.9: Esempio di revisione del budget durante l'anno

- Una voce di spesa può movimentare uno o più conti economici del bilancio. Quando si verifica questa circostanza il costo previsto viene suddiviso sui conti economici relativi, ogni riga riporterà il proprio importo.

- Esistono due componenti temporali da tenere in considerazione: la prima indica il mese cui l'importo di spesa si riferisce, mentre la seconda l'anno di riferimento del piano di budget. L'anno a cui il budget si riferisce è reperibile all'interno della tabella dimensionale linea budget.

2.4.3 Schema del fatto impegni

La granularità del fatto è rappresentata dall'impegno di spesa che un dipartimento ha preso per la fornitura di un bene o servizio durante un mese dell'anno di riferimento e del relativo conto economico che il costo movimentata.

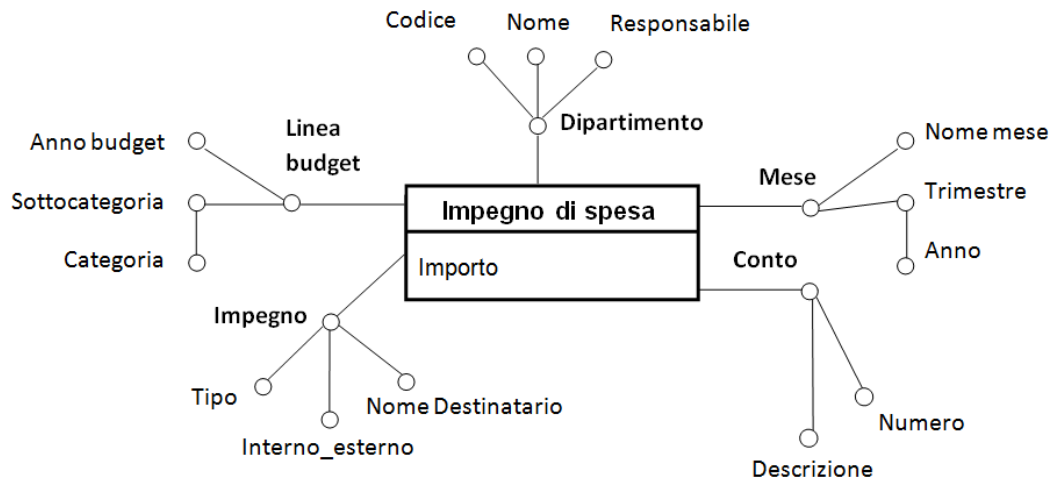


Figura 2.10: Schema concettuale del data mart degli impegni

Analizzando lo schema concettuale del data mart riportato in Figura 2.10 si possono fare alcune considerazioni.

- Un impegno può essere associato ad una o più linee del budget. In questa seconda circostanza l'impegno viene suddiviso su tutte le linee del budget riferite, in modo analogo a come avveniva nello schema del budget per le voci di spesa che movimentavano più conti economici.
- Esiste una dimensione aggiuntiva rispetto allo schema precedente, che si chiama impegno. Questa dimensione raccoglie le informazioni relative all'impegno preso:
 - Il tipo di impegno.

- Il destinatario dell’impegno, se interno o esterno all’azienda.
 - Il nome del destinatario.
- Confrontando budget e impegni, i manager possono tenere traccia del livello di spesa durante il corso dell’anno e rispondere a domande del tipo “Quanto ancora possiamo spendere?”.

2.4.4 Schema del fatto pagamenti

La granularità del fatto è rappresentata dalla fuoriuscita monetaria che un dipartimento ha in conseguenza della fornitura di un bene o servizio riferita ad un mese dell’anno di riferimento e del relativo conto economico che la spesa movimentata. Questa modellazione non interessa molto i manager aziendali che programmano il budget, bensì riveste importanza per l’area contabile-amministrativa.

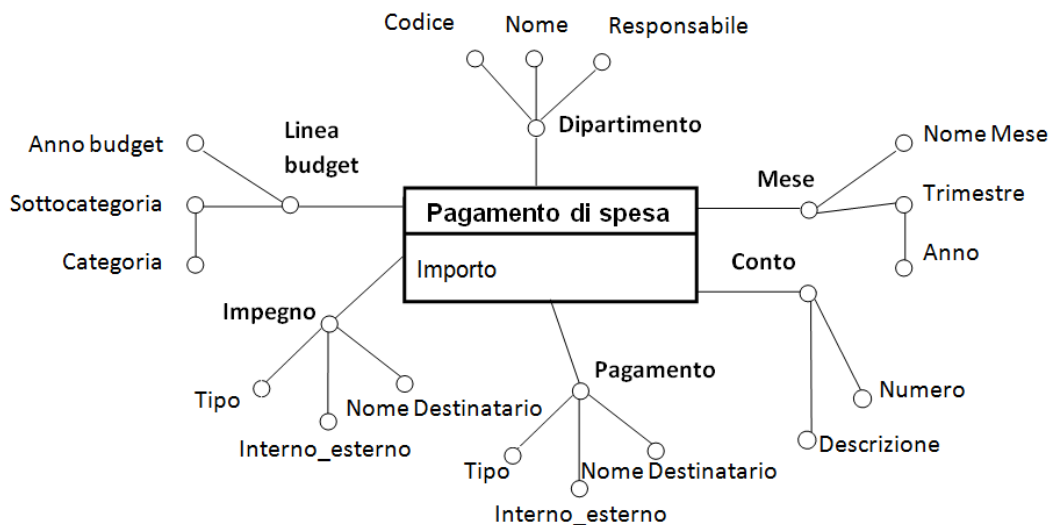


Figura 2.11: Schema concettuale del data mart dei pagamenti

Anche in questo caso, analizzando lo schema concettuale del data mart riportato in Figura 2.11 si possono fare alcune considerazioni importanti.

- Analogamente a quanto accaduto in precedenza, anche in questa circostanza un pagamento può essere associato ad uno o più impegni, seguendo la stessa modalità.

- Esiste una dimensione aggiuntiva rispetto allo schema concettuale degli impegni, che si chiama pagamento. Questa dimensione raccoglie le informazioni relative al pagamento effettuato:
 - Il tipo di pagamento.
 - Il destinatario del pagamento, se interno o esterno all'azienda.
 - Il nominativo dell'effettivo beneficiario.
- Un pagamento può essere associato oltre che agli impegni, direttamente alle corrispettive linee del budget.

2.4.5 Osservazioni conclusive

Il processo di budget non termina analizzando il flusso di cassa in uscita dell'azienda (i costi), ma prosegue studiando anche la parte riguardante le entrate (i ricavi). Analogamente a quanto visto per i costi si implementano le medesime tecniche per i ricavi, definendo tre tabelle dei fatti:

- *Le previsioni*: rappresenta la pianificazione dei ricavi che l'azienda si aspetta di ottenere.
- *I redditi*: sono gli impegni di vendita venturi che l'azienda ha intrapreso con i propri clienti.
- *Gli incassi*: rappresentano le entrate monetarie incassate dall'azienda a seguito della vendita di un servizio o prodotto.

Lo schema dei ricavi combinato e relazionato con lo schema dei costi può essere adoperato dai manager per ottenere un quadro complessivo della situazione budgetaria.

Questo tipo di analisi basata sulla distinzione tra i budget, gli impegni e i pagamenti, trova applicazione soprattutto nelle realtà aziendali di una certa complessità, dove i flussi di cassa in entrata e in uscita sono numerosi e dove si effettuano piani budgetari per ogni area strategica di affari presente. Per il caso oggetto del nostro studio e per gli obiettivi posti al nostro lavoro, come esposto nel successivo Capitolo, nel quale tratteremo nel dettaglio le fasi della progettazione, risulta sufficiente attuare esclusivamente una distinzione tra gli importi a budget e quelli a consuntivo, tralasciando le due entità impegni e redditi.

2.5 La struttura di un report per il budget

Tipicamente le informazioni di sintesi di supporto alle decisioni, elaborate e raccolte come descritto nella fase progettuale precedente, sono rappresentabili in forme molto differenti da quelle osservate. Solitamente si utilizzano strumenti grafici che presentano tali informazioni in formati visivamente piacevoli e intuitivi a favore degli utenti che ne faranno uso. Nel complesso, l'insieme dei documenti informativi, va a far parte della struttura del reporting [Marchi 08].

Nel senso più ampio del termine, il sistema dei rapporti informativi aziendali, più brevemente reporting, esprime l'insieme delle informazioni variamente rappresentate su supporti cartacei ed informatici, prodotte con periodicità e contenuto variabile, destinate alla comunicazione sia interna che esterna. In senso stretto invece, per sistema di reporting si intende un insieme più ristretto di informazioni, tipicamente quelle prodotte e destinate alla sola comunicazione interna per il controllo di gestione, rappresentabili in forma documentale e riferite a variabili chiave di controllo, su base comparativa e con l'evidenza dei valori ritenuti rilevanti.

Un sistema di reporting deve essere prima di tutto *efficace* e questa caratteristica dipende generalmente da tutti quegli elementi che ne definiscono la struttura: l'*articolazione* del rapporto su documenti elementari, la *rilevanza* e *selettività* dei dati, la *tempestività* del rapporto, l'*accuratezza* delle informazioni e la *flessibilità* e *accettabilità* del sistema.¹

Definire quindi un sistema di reporting efficace comporta la risoluzione delle problematiche inerenti a molteplici aspetti, tra cui quello tecnico-contabile e quello tecnico-informatico. Nel primo generalmente si va incontro ai limiti rappresentati dall'impossibilità o alla difficoltà di esprimere quantitativamente, o contabilmente, le manifestazioni di alcuni fenomeni. Nell'aspetto tecnico-informatico invece i limiti sono spesso legati alla tecnologia disponibile per il trattamento automatico delle informazioni, alla gestione delle basi di dati utilizzate e all'utilizzo dei sistemi di comunicazione prescelti. Anche l'aspetto comunicativo di un report è soggetto a problematiche

¹Per maggiori dettagli sul significato e le caratteristiche di ciascuna delle proprietà riportate, si rimanda il lettore alla consultazione del testo citato riportato in bibliografia.

differenti; tra le varie si incontrano difficoltà dovute alla ricerca di: capacità segnaletica di un report rispetto al destinatario, capacità di quest'ultimi di percepire il significato dei dati e della simbologia utilizzata nei messaggi, capacità di organizzare i contenuti dei report in modo differenziato in relazione alle diverse categorie di destinatari coinvolti (utilizzando diversi livelli di dettaglio e facendo ricorso a maggiore o minore uso di linguaggio amministrativo-contabile).

In relazione ai contenuti informativi dei documenti di reporting è possibile definire una classificazione che li distingue sulla base della tipologia di informazione riportata. A tale proposito si possono avere: *report orientati alla comunicazione interna o esterna, report quantitativi o descrittivi, report basati su fonti informative interne o esterne, report generali o analitici, report di pianificazione o controllo direzionale*. Tra questi diversi modi di definire un report, quello che maggiormente ha interessato la fase conclusiva di questo progetto è il report di *controllo direzionale*, dove si tiene conto delle finalità manageriali di definizione degli scenari futuri, degli obiettivi e di verifica dei risultati raggiunti.

Tipicamente i rapporti sono di pianificazione se contengono previsioni, simulazioni su ipotesi obiettivo e obiettivi attribuibili alle persone; mentre sono di controllo quando riportano analisi comparative dei dati, tipicamente tra i valori a budget e consuntivo, analisi degli scostamenti sugli obiettivi prefissati e ulteriori analisi dei risultati su base spaziale e temporale.

Nel complesso essi costituiscono una sintesi del sistema di contabilità direzionale d'impresa e contengono dati quantitativo-monetari fondamentali per il processo di controllo. Pertanto, il loro scopo fondamentale è quello di riuscire a fornire esaurienti risposte alle tipiche domande riguardanti i risultati delle attività svolte, sulle problematiche intrinseche nello svolgimento delle stesse e sulle tendenze in atto. A tale scopo un report di controllo direzionale necessita: di essere incentrato sulle variabili chiave controllabili, di riportare le informazioni su base comparativa e di evidenziare eventuali eccezioni o tendenze anomale sui dati. Un tipico esempio di report di controllo direzionale, mostrato in Figura 2.12, è quello dell'analisi comparativa su base mensile del fatturato a budget e a consuntivo riferito all'area commerciale. Il modello riportato è un tipico modulo generalmente predisposto con periodi-

cità mensile e destinato ai dirigenti commerciali. A partire da questo modello se ne possono poi costruire di più analitici, ad esempio con lo sviluppo del fatturato per area di vendita, per linee di prodotto o ancora per agenti di vendita, di norma predisposti al soddisfacimento delle esigenze conoscitive dei vari responsabili di area interessati.

MESE	CONSUNTIVO		BUDGET		SCOSTAMENTO			
	ANNO 2011							
	parziale	progressivo	parziale	progressivo	parziale	% su budget	progressivo	%su budget
Gennaio	7.828 €	7.828 €	8.425 €	8.425 €	-597 €	-7,1	-597 €	-7,1
Febbraio	8.141 €	15.696 €	8.256 €	16.681 €	-115 €	-1,4	-712 €	-4,3
Marzo	9.735 €	18.374 €	9.533 €	26.214 €	202 €	2,1	-510 €	-1,9
Aprile	9.627 €	35.841 €
Maggio	9.030 €	44.871 €
Giugno	10.156 €	55.027 €
Luglio	10.415 €	65.442 €
Agosto	7.554 €	72.996 €
Settembre	10.721 €	83.717 €
Ottobre	11.836 €	95.553 €
Novembre	12.060 €	107.613 €
Dicembre	14.282 €	121.895 €

Figura 2.12: Esempio di report di controllo direzionale

L'esempio mostrato è articolato per colonne mantenenti i dati interessanti distinguibili tra budget e consuntivo. I dati sono poi scissi tra valori parziali, utili all'analisi dell'andamento aziendale e degli scostamenti per ciascun mese, e tra valori progressivi, che invece presentano le stesse analisi in modo cumulativo.

Il calcolo dello scostamento è presentato sia in valore monetario che in valore percentuale sul budget (positivo se il valore totale a consuntivo è maggiore di quello a budget e negativo viceversa). Un altro aspetto interessante riguarda la cadenza mensile di generazione del report, che in questo caso si presume essere stato prodotto nel mese di aprile ancora in corso, in quanto i dati consuntivi non sono ancora stati rilevati, mentre gli importi totali a budget per mese sono già inseriti nel sistema di analisi.

2.6 Riassunto

- Si presenta una breve introduzione che ha per oggetto il budget, la sua utilità e la sua efficacia all'interno di un sistema azienda, mostrandone

- le varie tipologie esistenti, le aree ed i settori a cui questi sono riferiti.
- Tra le molteplici distinzioni osservate si dedica maggiore approfondimento al budget di tipo commerciale inteso come il budget definito allo scopo di esprimere gli obiettivi e i programmi dell'area commerciale in termini di volumi di vendita e conseguenti ricavi, considerandone i costi legati alla commercializzazione dei prodotti e/o servizi.
 - Sono mostrate le tecniche, le metodologie e le formule matematiche utilizzate per la determinazione del punto di pareggio, nel caso specifico inteso come l'uguaglianza tra l'ammontare dei costi totali aziendali (sia variabili che fissi) e i ricavi totali scaturiti dall'attività commerciale di vendita dei prodotti e/o servizi.
 - Viene riportato un modello di progettazione delle strutture dati, adeguate per la gestione di un piano di budget per l'azienda. Questo esempio è ampiamente trattato in letteratura [Adamson 98] e viene riportato al fine di osservare e studiare le problematiche inerenti il tema in oggetto, oltre ad ottenere delle solide linee guida verso la progettazione di un data warehouse che sarà discusso nel seguito.
 - L'esempio mostra i flussi informativi che interessano l'intero processo di budget, le singole informazioni che necessariamente devono essere presenti nel sistema e riporta i vari schemi concettuali dei data mart che modellano i processi di interesse.
 - Si introduce il tema relativo alla struttura di visualizzazione maggiormente utilizzata per i rapporti che contengono informazioni di budget. Si mostra un tipico esempio di come questi devono essere rappresentati.

Capitolo 3

Progettazione del data warehouse

Nel presente Capitolo si trattano tutte le tematiche inerenti la progettazione del data warehouse utilizzando una metodologia organizzata in fasi. Si procede iniziando con la presentazione del problema, passando poi alla fase di analisi dei requisiti, fino ad arrivare alla progettazione concettuale vera e propria, mostrando la trasformazione del modello concettuale in quello logico [Albano 09a].

3.1 Definizione del problema e analisi dei processi

Il committente del nostro lavoro, che per motivi di privacy non verrà menzionato, è un'azienda che prevalentemente fornisce consulenza informatica a numerose imprese dislocate sul territorio internazionale; ma si occupa anche della vendita di software gestionali modulari e soluzioni infrastrutturali. Tra i suoi clienti si riscontrano istituti di credito e società che sviluppano il proprio volume di affari prettamente nel campo sanitario-farmacologico, oltre a numerosi enti della pubblica amministrazione.

Attualmente il committente gestisce le informazioni inerenti la gestione economica e finanziaria delle propria attività attraverso l'impiego di differenti piattaforme non comunicanti fra loro:

- Un sistema CRM operativo per le vendite effettuate dai propri agenti.¹
- Un sistema gestionale per il monitoraggio dei costi avvenuti durante l'anno.
- Un applicativo per la stesura del budget annuale, sia per quanto concerne i costi che i ricavi.

Il committente ha esplicitato le necessità di realizzare uno strumento per la gestione delle vendite inerente i propri agenti recuperando le informazioni dalle rispettive basi di dati transazionali degli applicativi gestionali di cui sopra, per poter monitorare costantemente le attività lavorative della forza vendita, presentandole in termini di importi a consuntivo ed a budget. Per raggiungere l'obiettivo preposto si è reso quindi necessario definire e realizzare un data warehouse di supporto che permetta di effettuare le analisi dei ricavi derivanti dalle vendite e dei costi generati dall'intera gestione aziendale. Questo garantirà tempestive azioni correttive da parte del management qualora il sistema riveli situazioni di scostamento tra le due grandezze.

Ai fini di una corretta comprensione delle successive fasi di progettazione, occorre sottolineare alcuni punti fondamentali del problema affrontato:

- Attualmente l'azienda stila due piani di budget distinti; il primo inerente le previsioni dei costi dell'intera gestione aziendale, mentre il secondo inerente i ricavi derivanti dall'attività di vendita tramite gli agenti. Il management ha espressamente richiesto che questa separazione sia mantenuta anche nel nuovo sistema di budgeting. Il motivo principale che determina questa scelta sta nella struttura dell'azienda stessa, che distingue le unità organizzative in *centri di costo* e *centri di ricavo*, ciascuno dotato di un team formato da personale tecnico amministrativo e manager differenti.²

¹Descritto nella Sezione 7.2 - "SugarCRM - Il Customer Relationship Management".

²Da *Corso di economia aziendale - Vol.I - Modelli interpretativi aziendali*, a cura di P.Miolo Vitali, si riporta quanto segue: "in dottrina, solitamente si usa distinguere i *centri di responsabilità* tra *centri di costo* (poiché nell'area considerata è possibile stabilire una relazione diretta tra risorse impiegate e risultati ottenuti ed il costo sostenuto rappresenta l'elemento rilevante da tenere sotto controllo, pertanto si attribuiscono all'area obiettivi di efficienza in termini di costi sostenuti) e *centri di ricavo* (le leve decisionali sono rappresen-

- I due budget vengono redatti dai responsabili incaricati a termine dell'anno corrente, tipicamente nell'ultimo trimestre dell'anno solare, per l'anno successivo. Non sono soggetti a possibili revisioni quindi una volta definito un piano di budget, esso verrà mantenuto per tutto l'anno. Questo dettaglio potrebbe dare spunto a eventuali ampliamenti e miglioramenti futuri del sistema di budgeting.
- Esiste un'unica versione del budget, approvata in sede di consiglio di amministrazione aziendale ed inserita nel sistema dal personale incaricato. Anche se, in realtà, lo sviluppo di un piano di budget annuale richiede la stesura di molteplici versioni, solo per una di queste (tipicamente quella ritenuta più adeguata alle esigenze e agli obiettivi aziendali) se ne tiene traccia all'interno del sistema di gestione del budget.
- L'azienda non è organizzata in punti vendita presso i quali la clientela si può rivolgere e tutto ciò che costituisce un utile o un ricavo per l'organizzazione deriva esclusivamente dall'attività lavorativa degli agenti presso i clienti. Tali attività si distinguono prevalentemente in installazioni e manutenzioni di componenti hardware/software e consulenze informatiche. Ne deriva che ciascun agente, dotato di un proprio portafoglio clienti, costituisce il legame tra l'azienda e il cliente. Sarà suo compito curare e gestire tutte le richieste e le aspettative attese dai clienti una volta che questi hanno preso contatto con l'azienda.
- Ogni vendita effettuata è associata ad un unico agente.
- Ogni agente fa riferimento ad una sola sede aziendale, la quale ha un solo responsabile a capo.
- Da quanto detto assume particolare rilievo l'organizzazione di un piano di budget dei ricavi per ciascun agente. Interessa sapere quale tipo di transazione ha generato un determinato utile, se ad esempio deriva dalla vendita e installazione di particolari prodotti (e quali prodotti) o da eventuali servizi di consulenza erogati.

tate da elementi che influenzano il volume di ricavi da raggiungere)”. Ulteriori distinzioni riguardano i *centri di spesa*, *centri di profitto* o *di risultato* e *centri di investimento*.

- Per quanto riguarda le informazioni a consuntivo, i dati rilevanti vengono inseriti nelle corrispondenti basi di dati (descritte precedentemente) nel momento in cui l'informazione si manifesta.
- I dati di consuntivo, con cadenza mensile, sono forniti dal committente per aggiornare il data warehouse.
- In sede di analisi, la frequenza temporale con cui i dati vengono valutati ha una cadenza prevalentemente mensile. Ulteriori livelli di dettaglio utili sono rappresentati dal trimestre, il semestre e naturalmente l'anno corrente.
- I costi vengono classificati in tre categorie: i costi direttamente associati ad un agente, alla sede aziendale e quelli riferiti a personale non rientrante nella forza vendita. Inoltre sono suddivisi in costi fissi e variabili.

Alla luce delle richieste e considerazioni effettuate, si è pensato di procedere verso lo sviluppo di due data mart distinti, l'uno per il mantenimento dei dati relativi ai costi e l'altro per i ricavi. Considerando che lo stesso dato dovrà essere rappresentato nella sua forma consuntivata ed a budget, i data mart definiti saranno quattro: costi a budget, ricavi a budget, costi a consuntivo e ricavi a consuntivo. Questo ha permesso da un lato di poter adempiere alle richieste del committente e dall'altro di poter far fronte all'obiettivo principale che sta nella generazione del valore di scostamento tra budget e consuntivo per le singole voci, che verrà prodotto in fase di analisi. Il motivo fondamentale che però ha portato a questa scissione su data mart distinti degli importi preventivati e consuntivati risiede nel fatto che le informazioni che li popoleranno verranno generate da flussi ETL distinguibili per la fonte di dati in input.³ Essendo le informazioni recuperate da fonti differenti ed eterogenee tra loro, capita che queste presentino livelli di dettaglio differenti per la stessa entità: il caso più evidente che si verifica è l'assenza nel sistema gestionale inerente al budget del concetto di cliente, che invece risulta disponibile all'interno del CRM operativo. In fase di progettazione del data warehouse bisognerà tenere conto di queste divergenze.

Una volta raggiunto l'obiettivo principale si è poi presentata la possibilità di ampliare le prospettive di analisi dei dati contenuti nel data warehouse

³Di cui si discuterà ampiamente nel Capitolo 4.

tramite lo studio del punto di pareggio tra costi e ricavi.⁴ Per arrivare a questo si è resa indispensabile una fase aggiuntiva che ha permesso di conoscere approfonditamente le tecniche e le metodologie utilizzate dai vertici aziendali per la determinazione del proprio processo di budgeting. Solo in questo modo si è potuto riprogettare il processo il più conforme possibile alle regole descritte nel Capitolo precedente ridefinendo e riorganizzando le molteplici voci di costo tra costi fissi e variabili, diretti e indiretti.⁵ Per concludere, nelle Sezioni successive, si osserva la progettazione dei quattro data mart che rappresentano gli altrettanti processi definiti. Si distinguono nel processo di raccolta dei costi e processo dei ricavi; i primi derivanti principalmente da tutta la gestione aziendale e quindi anche dall'attività lavorativa degli agenti di vendita, i secondi derivanti soprattutto dalle sole attività di vendita e consulenza effettuate dagli agenti preposti. Ciascuno dei due processi si può ulteriormente distinguere tra la generazione di valori a consuntivo, quelli effettivamente riscontrati dalla gestione aziendale, ed a budget, quelli che invece vengono preventivati a termine dell'anno corrente per l'anno successivo.

3.2 Progettazione del data mart dei costi a budget

3.2.1 Raccolta dei requisiti di analisi

Si affronta ora la fase di raccolta e definizione dei requisiti delle analisi dei dati che in accordo con il committente si desidera eseguire per il fatto *costo a budget*. I requisiti più significativi sono presentati in Tabella 3.1.

1	Importo totale della voce di costo a budget per sede e per agente, per mese (trimestre, semestre e per l'intero anno corrente)
2	Totale importo a budget delle voci di costo per categoria e per tipologia
3	Importo totale del costo a budget per tutte le sedi

Tabella 3.1: Requisiti di analisi del processo costo a budget

⁴Si rimanda il lettore alla Sezione 2.3 - "Determinazione del punto di pareggio".

⁵Sezione 2.2 - "Il Budget commerciale".

3.2.2 Specifica dei requisiti

Preso in esame il processo *costo a budget* per il quale sono stati definiti i principali requisiti di analisi, si produce una specifica delle analisi dei dati in grado di evidenziare le dimensioni con gli attributi utili, le misure richieste e la granularità dei fatti.

N	Requisito di analisi	Dimensioni	Misure	Aggregazioni
1	Importo totale della voce di costo a budget per sede e per agente, per mese (trimestre, semestre e per l'intero anno corrente)	Agente, Sede, Voce Costo, Mese(trimestre, semestre, anno)	Importo budget	Somma(Importo budget)
2	Totale importo a budget delle voci di costo per categoria e per tipologia	Voce Costo(categoria, tipologia)	Importo budget	Somma(Importo budget)
3	Importo totale del costo a budget per tutte le sedi	Sede	Importo budget	Somma(Importo budget)

Per quanto riguarda la granularità del fatto, ai fini della soddisfazione del committente, interessa l'importo mensile di ogni singola voce di costo a budget che l'azienda dovrà sostenere.

Fatto budget costo

Descrizione	Dimensioni preliminari	Misure preliminari
Un fatto riguarda l'importo a budget di una voce di costo	Agente, Sede, Voce Costo, Mese	Importo budget

Si descrivono ora le dimensioni di analisi con i relativi attributi e le misure del fatto *budget costo*.

Dimensioni

Nome	Descrizione	Granularità
Agente	Definisce il singolo agente di vendita appartenente alla forza vendita totale dell'azienda	Un agente

Sede	Definisce la singola sede dell'intera azienda	Una sede
Voce Costo	Definisce la singola voce di costo a budget che l'azienda è tenuta a sostenere	Una voce di costo
Mese	Definisce il fronte temporale a cui si riferisce la singola voce di costo a budget	Un mese

Agente

Attributo	Descrizione	Tipo
Nome	Nome di battesimo dell'agente di vendita	Stringa
Telefono	Contatto telefonico dell'agente di vendita	Stringa
Email	Contatto di posta elettronica dell'agente di vendita	Stringa
Sede	Sede di riferimento dell'agente di vendita	Stringa

Sede

Attributo	Descrizione	Tipo
Nome	Nome della città in cui si colloca la sede	Stringa
Responsabile	Nome e Cognome del responsabile della sede	Stringa
Telefono	Contatto telefonico per quella sede	Stringa
Indirizzo	Indirizzo della sede	Stringa

Voce Costo

Attributo	Descrizione	Tipo
Nome	Nome univoco della voce di costo	Stringa
Categoria	Categoria in cui si colloca la voce di costo	Stringa
Tipo	Tipologia della voce di costo	Stringa
Descrizione	Breve descrizione testuale della voce di costo	Stringa

Mese

Attributo	Descrizione	Tipo
Nome mese	Nome del mese di riferimento	Stringa
Trimestre	Numero del trimestre di riferimento	Intero
Semestre	Numero del semestre di riferimento	Intero

Anno	Anno di riferimento	Intero
------	---------------------	--------

Misure

Misura	Descrizione	Aggregabilità	Derivata
Importo budget	Valore del costo preventivato a budget	Additiva	No

3.2.3 Progettazione concettuale del data mart

Dalla specifica dei requisiti appena discussa si è messo in evidenza il fatto interessante: *budget costo*. Gli attributi descritti nell'analisi dei dati danno spunto a un possibile schema concettuale iniziale del data mart, di cui si riporta la rappresentazione in Figura 3.1. Per la sua rappresentazione si è scelto di utilizzare il *Dimensional Fact Model* (il modello dimensionale dei fatti).

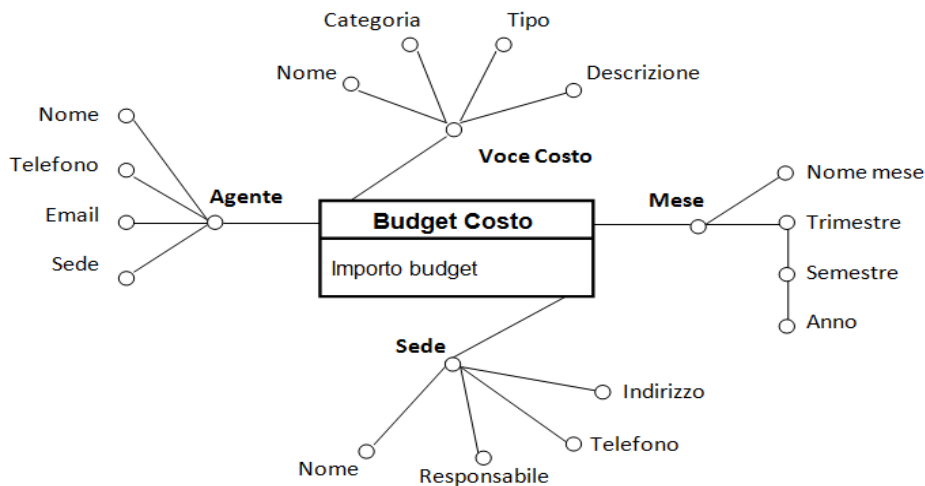


Figura 3.1: Schema concettuale iniziale del data mart *budget costo*

3.3 Progettazione del data mart dei ricavi a budget

3.3.1 Raccolta dei requisiti di analisi

In questa Sezione si affronta nel dettaglio la fase di raccolta e definizione dei requisiti delle analisi dei dati che in accordo con il committente si desidera eseguire per il fatto *ricavo a budget*. Alcuni dei requisiti di analisi più significativi sono presentati in Tabella 3.2.

4	Totale del ricavo di vendita preventivato per agente, per tipologia di ricavo, per mese (per anno)
5	Ammontare complessivo dell'importo ricavato a preventivo di un agente per trimestre e semestre per tutti i prodotti o servizi
6	Ricavo a budget per sede, per l'anno corrente

Tabella 3.2: Requisiti di analisi del processo ricavo a budget

3.3.2 Specifica dei requisiti

Preso in esame il processo *budget del ricavo* per il quale sono stati definiti i principali requisiti di analisi, si produce una specifica delle analisi dei dati in grado di evidenziare le dimensioni con gli attributi utili, le misure richieste e la granularità dei fatti.

N	Requisito di analisi	Dimensioni	Misure	Aggregazioni
4	Totale del ricavo di vendita preventivato per agente, per tipologia di ricavo, per mese (per anno)	Agente, Tipo Ricavo, Mese(anno)	Importo budget	Somma(Importo budget)
5	Ammontare complessivo dell'importo ricavato a preventivo di un agente per trimestre e semestre per tutti i prodotti o servizi	Agente, Prodotto, Mese(trimestre, semestre)	Importo budget	Somma(Importo budget)

6	Ricavo a budget per sede, per l'anno corrente	Sede, Mese(anno)	Importo budget	Somma(Importo budget)
---	---	---------------------	----------------	-----------------------

Per quanto riguarda la granularità del fatto, ai fini della soddisfazione del committente, interessa l'importo mensile della singola voce di ricavo a budget che l'azienda prevede di incassare in cambio di determinate prestazioni offerte.

Fatto budget ricavo

Descrizione	Dimensioni preliminari	Misure preliminari
Un fatto riguarda l'importo preventivato di una voce di ricavo derivante da una prestazione futura	Agente, Prodotto, Tipo Ricavo, Mese	Importo budget

Si descrivono ora le dimensioni di analisi con i relativi attributi e le misure del fatto *budget ricavo*.

Dimensioni

Nome	Descrizione	Granularità
Agente	Definisce il singolo agente di vendita appartenente alla forza vendita totale dell'azienda	Un agente
Tipo Ricavo	Definisce se un ricavo deriva da una vendita di prodotti o da una erogazioni di servizi o consulenze	Una tipologia
Prodotto	Definisce un prodotto del portafoglio aziendale	Un prodotto
Mese	Definisce il fronte temporale a cui si riferisce la singola voce di ricavo consuntivata	Un mese

Agente

Attributo	Descrizione	Tipo
Nome	Nome di battesimo dell'agente di vendita	Stringa
Telefono	Contatto telefonico dell'agente di vendita	Stringa
Email	Contatto di posta elettronica dell'agente di vendita	Stringa
Sede	Sede di riferimento dell'agente di vendita	Stringa

Tipo Ricavo

Attributo	Descrizione	Tipo
Descrizione	Sintesi descrittiva della tipologia associabile ai prodotti	Stringa

Prodotto

Attributo	Descrizione	Tipo
Nome	Nome del prodotto in portafoglio	Stringa
Descrizione	Sintesi descrittiva delle caratteristiche del prodotto	Stringa

Mese

Attributo	Descrizione	Tipo
Nome mese	Nome del mese di riferimento	Stringa
Trimestre	Numero del trimestre di riferimento	Intero
Semestre	Numero del semestre di riferimento	Intero
Anno	Anno di riferimento	Intero

Misure

Misura	Descrizione	Aggregabilità	Derivata
Importo budget	Valore del ricavo preventivato a budget	Additiva	No

3.3.3 Progettazione concettuale del data mart

Dalla specifica dei requisiti appena discussa si è messo in evidenza il fatto interessante: *budget ricavo*. Gli attributi descritti nell'analisi dei dati danno spunto a un possibile schema concettuale iniziale del data mart, di cui si riporta la rappresentazione in Figura 3.2.

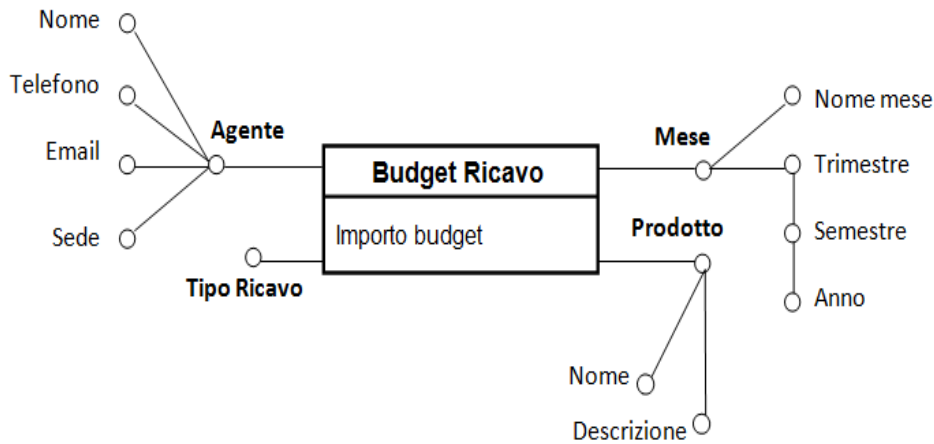


Figura 3.2: Schema concettuale iniziale del data mart *budget ricavo*

3.4 Progettazione del data mart dei costi a consuntivo

3.4.1 Raccolta dei requisiti di analisi

In questa Sezione si affronta nel dettaglio la fase di raccolta e definizione dei requisiti delle analisi dei dati che in accordo con il committente si desidera eseguire per il fatto *costo a consuntivo*. Alcuni dei requisiti di analisi più significativi sono presentati in Tabella 3.3.

7	Importo totale della voce di costo consuntivata per sede, per agente, per mese (trimestre, semestre e per l'intero anno corrente)
8	Totale importo a consuntivo delle voci di costo per categoria e per tipologia
9	Importo totale del costo consuntivato per tutte le sedi

Tabella 3.3: Requisiti di analisi del processo costo a consuntivo

3.4.2 Specifica dei requisiti

Preso in esame il processo *costo a consuntivo* per il quale sono stati definiti i principali requisiti di analisi, si produce una specifica delle analisi dei dati in grado di evidenziare le dimensioni con gli attributi utili, le misure richieste e

la granularità dei fatti.

N	Requisito di analisi	Dimensioni	Misure	Aggregazioni
7	Importo totale della voce di costo consuntivata per sede, per agente, per mese (trimestre, semestre e per l'intero anno corrente)	Agente, Sede, Voce Costo, Mese(trimestre, semestre, anno)	Importo consuntivo	Somma(Importo consuntivo)
8	Totale importo a consuntivo delle voci di costo per categoria e per tipologia	Voce Costo(categoria, tipologia)	Importo consuntivo	Somma(Importo consuntivo)
9	Importo totale del costo consuntivato per tutte le sedi	Sede	Importo consuntivo	Somma(Importo consuntivo)

Per quanto riguarda la granularità del fatto, ai fini della soddisfazione del committente, interessa l'importo mensile di ogni singola voce di costo a consuntivo che l'azienda è tenuta a sostenere.

Fatto costo a consuntivo

Descrizione	Dimensioni preliminari	Misure preliminari
Un fatto riguarda l'importo consuntivato di una voce di costo	Agente, Sede, Voce Costo, Mese	Importo consuntivo

Si descrivono ora le dimensioni di analisi con i relativi attributi e le misure del fatto *costo a consuntivo*.

Dimensioni

Nome	Descrizione	Granularità
Agente	Definisce il singolo agente di vendita appartenente alla forza vendita totale dell'azienda	Un agente
Sede	Definisce la singola sede dell'intera azienda	Una sede

Voce Costo	Definisce la singola voce di costo a consuntivo che l'azienda è tenuta a sostenere	Una voce di costo
Mese	Definisce il fronte temporale a cui si riferisce la singola voce di costo consuntivata	Un mese

Agente

Attributo	Descrizione	Tipo
Nome	Nome di battesimo dell'agente di vendita	Stringa
Telefono	Contatto telefonico dell'agente di vendita	Stringa
Email	Contatto di posta elettronica dell'agente di vendita	Stringa
Sede	Sede di riferimento dell'agente di vendita	Stringa

Sede

Attributo	Descrizione	Tipo
Nome	Nome della città in cui si colloca la sede	Stringa
Responsabile	Nome e Cognome del responsabile della sede	Stringa
Telefono	Contatto telefonico per quella sede	Stringa
Indirizzo	Indirizzo della sede	Stringa

Voce Costo

Attributo	Descrizione	Tipo
Nome	Nome univoco della voce di costo	Stringa
Categoria	Categoria in cui si colloca la voce di costo	Stringa
Tipo	Tipologia della voce di costo	Stringa
Descrizione	Breve descrizione testuale della voce di costo	Stringa

Mese

Attributo	Descrizione	Tipo
Nome mese	Nome del mese di riferimento	Stringa
Trimestre	Numero del trimestre di riferimento	Intero
Semestre	Numero del semestre di riferimento	Intero
Anno	Anno di riferimento	Intero

Misure

Misura	Descrizione	Aggregabilità	Derivata
Importo consuntivo	Valore del costo rilevato a consuntivo	Additiva	No

3.4.3 Progettazione concettuale del data mart

Dalla specifica dei requisiti appena discussa si è messo in evidenza il fatto interessante: *costo a consuntivo*. Gli attributi descritti nell'analisi dei dati danno spunto a un possibile schema concettuale iniziale del data mart, di cui si riporta la rappresentazione in Figura 3.3.

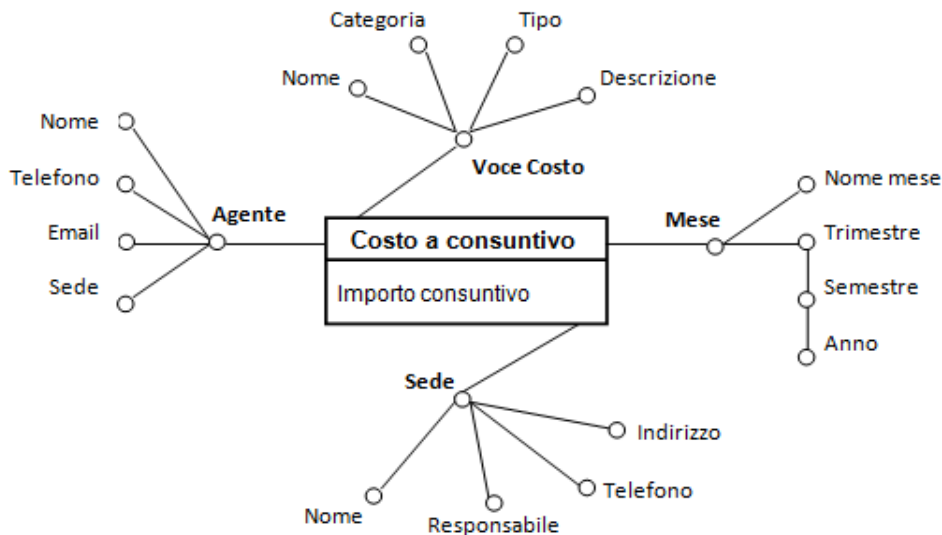


Figura 3.3: Schema concettuale iniziale del data mart *costo a consuntivo*

3.5 Progettazione del data mart dei ricavi a consuntivo

3.5.1 Raccolta dei requisiti di analisi

In questa Sezione si affronta nel dettaglio la fase di raccolta e definizione dei requisiti delle analisi dei dati che in accordo con il committente si desidera

eseguire per il fatto *ricavo a consuntivo*. Alcuni dei requisiti di analisi più significativi sono presentati in Tabella 3.4.

10	Ammontare complessivo del fatturato di un agente, per trimestre e per semestre
11	Totale del ricavo di vendita fatturato da un agente per prodotto, per mese (anno)
12	Importo del fatturato totale per sede, per l'anno corrente

Tabella 3.4: Requisiti di analisi del processo ricavo a consuntivo

3.5.2 Specifica dei requisiti

Preso in esame il processo *ricavo a consuntivo* per il quale sono stati definiti i principali requisiti di analisi, si produce una specifica delle analisi dei dati in grado di evidenziare le dimensioni con gli attributi utili, le misure richieste e la granularità dei fatti.

N	Requisito di analisi	Dimensioni	Misure	Aggregazioni
10	Ammontare complessivo del fatturato di un agente, per trimestre e per semestre	Agente, Mese(trimestre, semestre)	Importo consuntivo	Somma(Importo consuntivo)
11	Totale del ricavo di vendita fatturato da un agente per prodotto, per mese (anno)	Agente, Prodotto, Mese(anno)	Importo consuntivo	Somma(Importo consuntivo)
12	Importo del fatturato totale per sede, per l'anno corrente	Sede, Mese(anno)	Importo consuntivo	Somma(Importo consuntivo)

Per quanto riguarda la granularità del fatto, ai fini della soddisfazione del committente, interessa l'importo mensile di ogni singola voce di ricavo a consuntivo che l'azienda riceve in cambio di determinate prestazioni cedute. A seguire, come per le Sezioni precedenti, si descrivono le dimensioni di analisi con i relativi attributi e le misure del fatto *ricavo a consuntivo*.

Fatto ricavo a consuntivo

Descrizione	Dimensioni preliminari	Misure preliminari
Un fatto riguarda l'importo consuntivato di una voce di ricavo derivante da una prestazione ceduta	Agente, Prodotto, Mese	Importo consuntivo

Dimensioni

Nome	Descrizione	Granularità
Agente	Definisce il singolo agente di vendita appartenente alla forza vendita totale dell'azienda	Un agente
Prodotto	Definisce un prodotto del portafoglio aziendale	Un prodotto
Mese	Definisce il fronte temporale a cui si riferisce la singola voce di ricavo consuntivata	Un mese

Agente

Attributo	Descrizione	Tipo
Nome	Nome di battesimo dell'agente di vendita	Stringa
Telefono	Contatto telefonico dell'agente di vendita	Stringa
Email	Contatto di posta elettronica dell'agente di vendita	Stringa
Sede	Sede di riferimento dell'agente di vendita	Stringa

Prodotto

Attributo	Descrizione	Tipo
Nome	Nome del prodotto in portafoglio	Stringa
Descrizione	Sintesi descrittiva delle caratteristiche del prodotto	Stringa

Mese

Attributo	Descrizione	Tipo
Nome mese	Nome del mese di riferimento	Stringa

Trimestre	Numero del trimestre di riferimento	Intero
Semestre	Numero del semestre di riferimento	Intero
Anno	Anno di riferimento	Intero

Misure

Misura	Descrizione	Aggregabilità	Derivata
Importo consuntivo	Valore del ricavo rilevato a consuntivo	Additiva	No

3.5.3 Progettazione concettuale del data mart

Dalla specifica dei requisiti appena discussa si è messo in evidenza il fatto interessante: *ricavo a consuntivo*. Gli attributi descritti nell'analisi dei dati danno spunto a un possibile schema concettuale iniziale del data mart, di cui si riporta la rappresentazione in Figura 3.4.

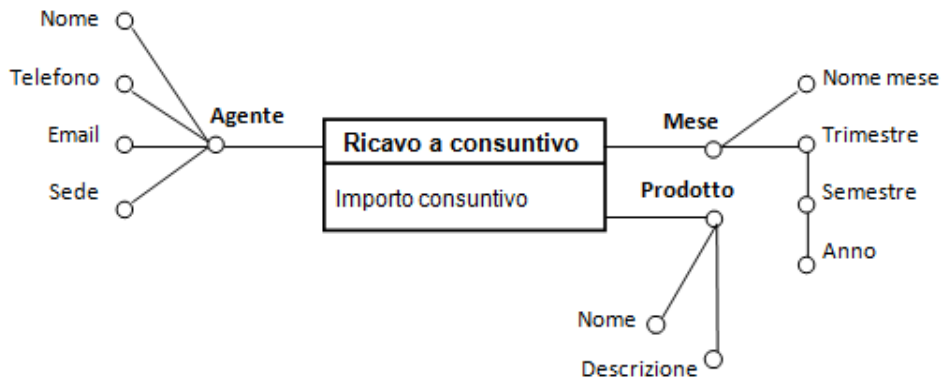


Figura 3.4: Schema concettuale iniziale del data mart *ricavo a consuntivo*

Presentate le specifiche dei requisiti per ciascun fatto, si rappresentano le relative dimensioni e misure nella seguente forma tabellare per mettere in evidenza quali dimensioni e quali misure possono essere in comune a fatti diversi e quindi uniformate o eventualmente ridenominate. Ad esempio le dimensioni *Mese* e *Agente* sono comuni a tutti i fatti e pertanto si assume di uniformarle utilizzando quelle del fatto *costo a consuntivo*.

Convenuto che la base di dati operativa è stata sviluppata solo in tempi ampiamente successivi rispetto alla progettazione del data warehouse, oggetto di discussione di questo Capitolo, la fase di *progettazione concettuale*

dei data mart dai dati operazionali non è stata affrontata; pertanto gli schemi concettuali iniziali appena presentati sono da considerarsi in qualità di schemi concettuali finali per i data mart.

Dimensioni dei fatti

Dimensione	Costo a consuntivo	Budget costo	Ricavo a consuntivo	Budget ricavo
Agente	X	X	X	X
Mese	X	X	X	X
Prodotto			X	X
Sede	X	X		
Tipo di ricavo				X
Voce di costo	X	X		

Misure dei fatti

Misura	Costo a consuntivo	Budget costo	Ricavo a consuntivo	Budget ricavo
Importo consuntivo	X			
Importo budget		X		
Importo consuntivo			X	
Importo budget				X

3.6 Progettazione logica dei data mart

Si affronta ora la generazione di schemi a stella per i data mart. Generalmente, per ciascuna tabella dei fatti definita durante la progettazione concettuale, si definisce una tabella dei fatti dello schema a stella, mentre per quanto riguarda le dimensioni, si definisce la corrispondente tabella in relazione con la tabella dei fatti. Queste operazioni sono eseguite definendo opportunamente le chiavi e le chiavi esterne.

È importante, in questa fase della progettazione, fare alcune distinzioni tra i potenziali schemi a stella che possono essere definiti a partire dalla progettazione logica appena presentata.

1. Si definisce un data mart per ciascun fatto interessante: *budget costo*, *budget ricavo*, *costo a consuntivo*, *ricavo a consuntivo*. In questo caso per ciascun data mart si mantiene l'*importo* come unica misura interessante.
2. Si definisce un data mart per ciascuna coppia di fatti interessanti: *costo budget e consuntivo*, *ricavo budget e consuntivo*. In questo caso per ciascun data mart si mantengono rispettivamente gli importi del *costo budget* e *costo consuntivo* nel primo caso e il *ricavo budget* e *ricavo consuntivo* nel secondo come misure interessanti, mentre il valore di *scostamento* può essere calcolato come differenza delle due misure precedenti in fase di analisi.
3. Si definisce un data mart per ciascuna coppia di fatti interessanti: *costo budget e consuntivo*, *ricavo budget e consuntivo*. A differenza della soluzione precedente, in questo caso per ciascun data mart si mantengono rispettivamente gli importi del *costo budget*, *costo consuntivo* e *scostamento* nel primo caso e il *ricavo budget*, *ricavo consuntivo* e *scostamento* nel secondo come misure interessanti.
4. Si definisce un unico data mart per tutti i fatti interessanti: *costo e ricavo a budget e consuntivo*. In questo caso si mantiene l'*importo* come unica misura e gli attributi *se budget o consuntivo*, *se costo o ricavo* per definire correttamente la misura.

La seconda soluzione risulta quella che meglio si conforma sia alle esigenze dell'azienda cliente che alle esigenze di sviluppo dello strumento di analisi. Questa soluzione infatti aggrega il vantaggio di mantenere una struttura snella che meglio si adatta alla separazione tra grandezze riferite ai costi e quelle riferite ai ricavi (come espresso in fase di raccolta dei requisiti), ma anche il vantaggio di avere una struttura che meglio si adegua agli strumenti analitici utilizzati (di cui si discuterà nel Capitolo conclusivo), che garantisca di massimizzare le prestazioni e di minimizzare i tempi di risposta del sistema ogni qual volta che viene interrogato. Rimandando alla Sezione successiva i vantaggi e svantaggi delle potenziali scelte ed i motivi che hanno spinto verso la seconda delle quattro soluzioni, si osservano ora gli schemi relazionali dei data mart riportati in Figura 3.5 e Figura 3.6.

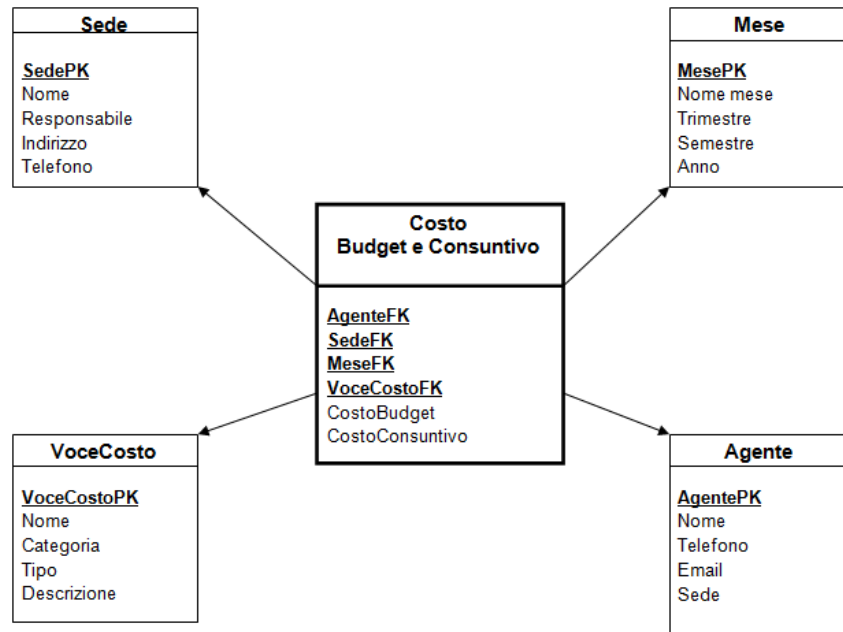


Figura 3.5: Schema a stella del data mart costo budget e consuntivo

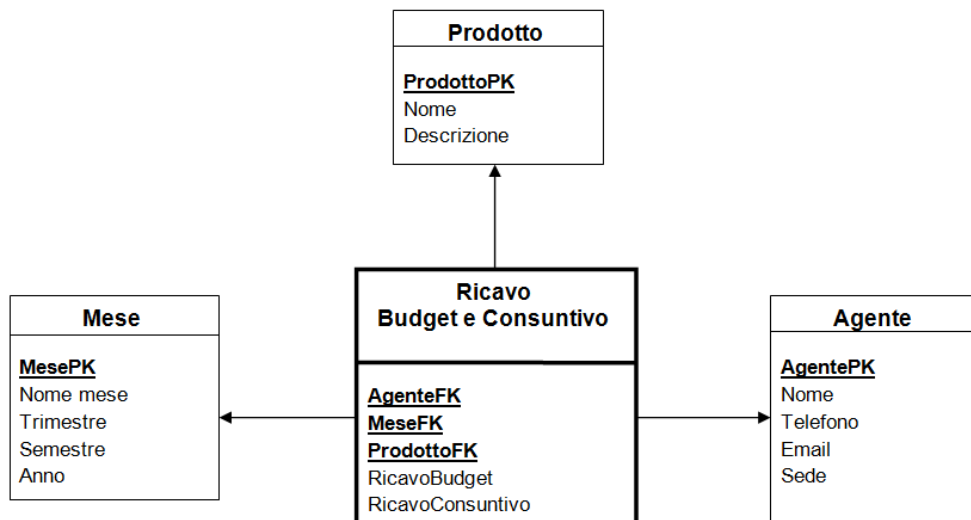


Figura 3.6: Schema a stella del data mart ricavo budget e consuntivo

Si nota come in Figura 3.6 siano mantenute solamente le tre tabelle dimensionali, ed i relativi attributi, comuni ai due fatti rilevati durante la progettazione concettuale dei data mart, tralasciando la dimensione *tipo ricavo* (che risulta essere una dimensione presente nel solo fatto *budget ricavo*). Questa scelta deriva da problematiche inerenti alle differenti granularità riscontrate

nei due fatti inizialmente rilevati. Inoltre, si evidenzia, come il committente non ha ritenuto interessante eseguire delle analisi incentrate anche sulla dimensione *cliente*, nonostante questa informazione sia presente in alcuni fatti, come ad esempio il *ricavo a consuntivo*. Questa decisione è maturata in accordo col committente in quanto lo studio è incentrato esclusivamente sull'entità agente. Lo schema a stella prescelto è mostrato in Figura 3.6.

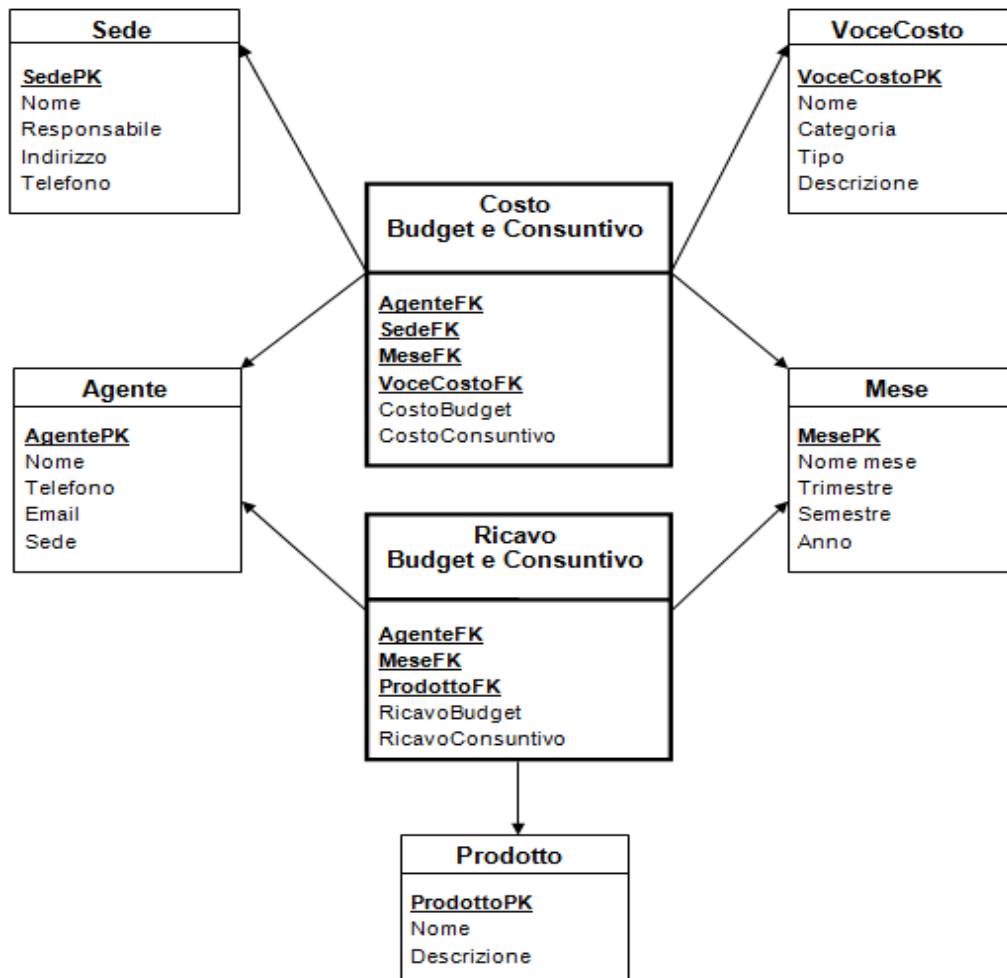


Figura 3.7: Schema a costellazione del data warehouse

Per concludere i due data mart sono poi integrati per poter definire lo schema del data warehouse. Questa integrazione è eseguita rispettando i vincoli di condivisione delle tabelle *agente* e *data*. Lo schema conclusivo è pertanto uno schema a costellazione in cui esistono due tabelle dei fatti che condividono

alcune dimensioni. La struttura completa del data warehouse è mostrata in Figura 3.7.

Si nota come, in questo schema finale, ciascuna tabella delle dimensioni, oltre agli attributi, mantiene anche la propria chiave primaria surrogata: *AgentePK*, *MesePK*, *ProdottoPK*, *SedePK* e *VoceCostoPK*. Per quanto riguarda invece le tabelle dei fatti, sono aggiunte le chiavi esterne riferite alle tabelle dimensionali. In particolare per il fatto *costo budget e consuntivo* la chiave esterna è costituita da: *AgenteFK*, *MeseFK*, *SedeFK* e *VoceCostoFK*, mentre per il fatto *ricavo budget e consuntivo* la chiave esterna è costituita da: *AgenteFK*, *MeseFK* e *ProdottoFK*.

3.7 Osservazioni conclusive

I motivi che hanno spinto verso la progettazione dei due data mart nelle modalità appena descritte possono essere ricondotti alle seguenti osservazioni:

- In primo luogo, si riporta nella Tabella 3.5 un confronto incrociato su alcune situazioni di vantaggio e svantaggio inerenti all'adozione di ciascuna delle soluzioni indicate nella Sezione precedente. Si riportano poi, nelle Figure 3.8, 3.9 e 3.10 i potenziali schemi delle tabelle dei fatti inerenti alle soluzioni alternative considerate, tralasciando le tabelle dimensionali che risultano pressoché invariate.
- La richiesta di mantenere separati i dati (quindi su data mart distinti) che costituiscono un costo aziendale da quelli che determinano un ricavo è stata una scelta presa dai responsabili dell'azienda cliente. Ciò dipende soprattutto dalle prospettive future che essi si aspettano dal data warehouse. Infatti, al termine di questo progetto, è previsto un ampliamento dello strumento per il controllo di gestione in ottica di studio delle marginalità. È importante a tale fine definire una netta distinzione di tutte o parte delle voci che costituiscono un costo di gestione aziendale, le quali dovranno poi essere ulteriormente identificabili come appartenenti alle classi dei costi fissi diretti o indiretti e costi variabili diretti o indiretti. Questo permetterà, utilizzando opportune

modalità di calcolo, di identificare alcuni tradizionali indici economici prestazionali [Cinquini 08] quali: *MOL* (Margine Operativo Lordo), *MC1* (Margine di Contribuzione di primo livello), *MC2* (Margine di Contribuzione di secondo livello), *UO* (Utile Operativo) ecc.

	Sol. 1	Sol. 2	Sol. 3	Sol. 4
Maggiore adeguatezza alle specifiche convenute in fase di progettazione	X	X	X	
Problematiche inerenti al caricamento dati tramite flussi ETL			X	X
Problematiche dovute all'utilizzo degli strumenti analitici che non permettono di fare analisi contemporaneamente su più data mart	X			
Problematiche inerenti all'utilizzo di tabelle dimensionali differenti che caratterizzano i ricavi a budget e consuntivo		X	X	X
Possibilità di calcolo della misura di scostamento in fase di analisi	X	X		
Massimizzazione dello spazio utilizzato nella memorizzazione delle informazioni per ciascun data mart			X	X
Forte complessità nella generazione di interrogazioni SQL				X
Minimizzazione dei tempi di risposta del sistema nella restituzione dell'output desiderato		X	X	

Tabella 3.5: Vantaggi e svantaggi per le possibili soluzioni

- Posto il maggior dettaglio sui dati che costituiscono informazioni di costo consuntivo, si è scelto di mantenere la stessa logica per i dati che identificano i ricavi aziendali. Dovendo poi realizzare un data warehouse per il controllo di gestione incentrato sulla stesura di un piano



Figura 3.8: Tabelle dei fatti nella soluzione 1

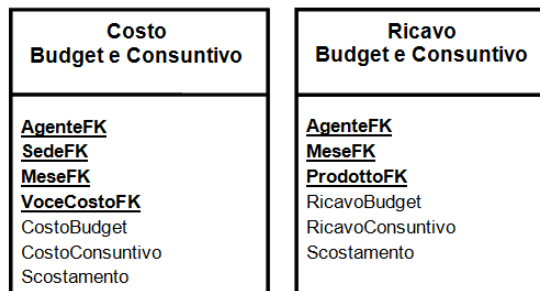


Figura 3.9: Tabelle dei fatti nella soluzione 3



Figura 3.10: Tabella dei fatti nella soluzione 4

di budget annuale la scelta è ricaduta pertanto sulla separazione delle informazioni nei due fatti fondamentali che definiscono costi e ricavi sia in termini di budget che a consuntivo.

- In merito alla numerosità delle informazioni che andranno a popolare i data mart della soluzione adottata, si riportano i valori nella Tabella 3.6. Tali valori si riferiscono a dati di prova approssimati che si discostano da quelli effettivi, dal momento che non è stato possibile popolare il data warehouse delle informazioni reali per motivi di privacy nel trattamento dati.

N° record	Tabelle dimensionali
10	Agente
12	Data
10	Prodotto
3	Sede
40	Voce di costo

Tabella 3.6: Numerosità approssimativa delle tabelle dimensionali

Preso in considerazione l'arco temporale di un anno, previsto per il piano di budget, distinto nei 12 mesi costituenti l'unità di rilevazione di un costo o di un ricavo e considerato che l'ammontare di record presenti nelle tabelle dei fatti è costituito dal numero delle possibili combinazioni di chiavi; posti i valori sopra indicati si ottengono approssimativamente 14.400 record per il fatto *costo budget e consuntivo* e circa 2.400 record per il fatto *ricavo budget e consuntivo*. Considerato poi, che in fase di progettazione e popolamento del data warehouse non è stato possibile sfruttare informazioni realmente generate dall'azienda cliente, sono state utilizzate informazioni di prova (dati di demo) appositamente definiti, per un ammontare pari a 2.000 record per il fatto *costo budget e consuntivo* e 500 record per il fatto *ricavo budget e consuntivo*.

Un altro tema che merita un approfondimento riguarda invece il trattamento delle dimensioni o attributi dimensionali *multivalore*.⁶ Anche se questa problematica interessa maggiormente la fase di progettazione della base di dati,

⁶In questo contesto il tema non è stato affrontato dal momento che non se ne sono presentate le necessità durante lo sviluppo dei data mart. Come già detto, una vendita e quindi un record di ricavo a consuntivo è associato ad un unico agente incaricato. Non

non tutte le possibili soluzioni mostrate in letteratura a riguardo hanno la stessa efficacia in termini di prestazioni di analisi sullo strumento Pentaho C.E. e ciò è dipeso da molteplici fattori.

Poniamo caso in cui una vendita (e quindi un ricavo a consuntivo) possa essere conseguita da più agenti, situazione rappresentata dal punto di vista logico in Figura 3.11.

Tabella del fatto VENDITE					
id_vendita	...	id_agente_1	id_agente_2	...	importo
...
100	...	1	3	...	1.000
101	...	1	2	...	2.000
...

Tabella dimensionale AGENTE				
id_agente	...	nome	cognome	...
1	...	Mario	Bianchi	...
2	...	Marco	Rossi	...
3	...	Giovanni	Verdi	...
...

Figura 3.11: Esempio di vendita multiagente

Esistono molteplici strategie e tecniche per risolvere questa problematica, riportate su [Albano 09a], tra cui quella che prevede la duplicazione dei dati. In altri termini, se una vendita può essere effettuata da più agenti (in questo esempio da solo due agenti), si può prevedere la duplicazione della stessa informazione per entrambi gli agenti, come mostrato logicamente in Figura 3.12.

Utilizzando la duplicazione dei dati però si evidenzia un'ulteriore problematica, quella della corretta ripartizione del valore della misura. Come mostrato in figura, la misura importo per la vendita 100 è stata riportata in maniera analoga per entrambi gli agenti. Ciò implica che in sede di analisi, se l'utente richiede l'ammontare complessivo scaturito da tale vendita (SUM(importo)), egli otterrà l'informazione errata di €2.000. Pertanto questa tecnica deve necessariamente essere accompagnata da particolari strategie di ripartizione delle informazioni (in questo caso informazioni di tipo monetario), che tipicamente sono regolamentate dalle metodologie di business reali

è previsto che due o più agenti possano partecipare allo stesso procedimento di vendita presso un cliente.

utilizzate dal committente.

Tabella del fatto VENDITE				
id_vendita	...	id_agente	...	importo
...
100	...	1	...	1.000
100	...	3	...	1.000
...

Figura 3.12: Esempio di duplicazione dei dati sulla tabella dei fatti

Ad esempio si possono prevedere ripartizioni percentuali diverse sull'ammontare complessivo, del tipo 50% per entrambi gli agenti oppure il 70% per l'agente con un più altro grado di anzianità e il restante 30% per l'altro e così via. Utilizzando lo strumento di BI Pentaho C.E., la scelta di questa alternativa alla risoluzione del problema è funzionale all'incremento del numero di record nella tabella dei fatti. Pentaho C.E., in quanto basato sul linguaggio *Java*, eredita i limiti legati all'utilizzo della *Java Virtual Machine*, quindi la soluzione migliore tipicamente ricade sull'alternativa che minimizza il numero di record. Allo scopo occorre conoscere anticipatamente il volume di dati che lo strumento dovrà gestire, quante volte le tabelle verranno aggiornate ed altri dettagli, fino ad eseguire dei calcoli previsionali di quante transazioni verranno inserite nell'arco di un anno, quanti anni si ritiene opportuno che il database dovrà mantenere, quale sarà il livello di dettaglio cui tener conto ecc. Ad esempio, supponendo che l'azienda cliente esegue mediamente 1.000.000 di vendite all'anno, che ciascuna vendita è seguita da almeno 3 agenti e che il sistema è tenuto a mantenere i dati degli ultimi 10 anni si raggiunge un volume di dati medio di circa 30.000.000 di record.

Dal momento che nel caso di studio riportato in questa tesi, il volume di dati previsto ricade sull'ordine delle decine di migliaia di record, la tecnica della duplicazione delle informazioni può essere un'alternativa valida alla risoluzione del problema.

Successivamente, fatti questi studi previsionali, è opportuno anche tenere in considerazione il numero degli utenti che hanno accesso al sistema, un altro fattore che può influire sulle scelte progettuali.

Se per le motivazioni appena citate, la duplicazione dei record non risulta un'alternativa valida, si possono implementare ulteriori soluzioni. Tra queste la possibilità di avvalersi di una *bridge table*; una tabella ausiliaria che si

frappone tra la tabella dei fatti vendite e quella della dimensione agente, mantenente l'associazione di chiavi esterne tra l'agente e la vendita, come mostrato in Figura 3.13.

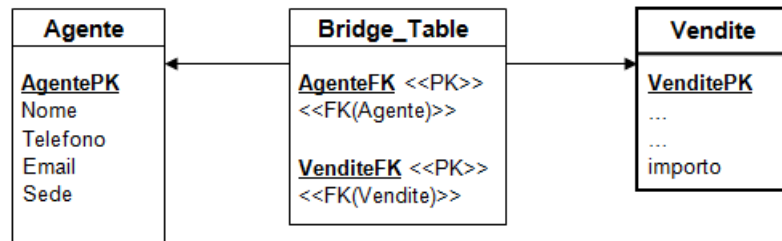


Figura 3.13: Esempio di utilizzo della bridge table

Anche se questa soluzione risulta valida ai fini risolutivi del problema, essa però viola le proprietà di uno schema a stella e quindi, per le motivazioni che verranno discusse nei Capitoli successivi a riguardo dell'utilizzo di uno schema a stella per le analisi con la piattaforma Pentaho C.E., è preferibile non utilizzarla.

Un'altra valida alternativa è mostrata in Figura 3.14. In questo caso si sfrutta l'utilizzo di una nuova tabella che mantiene l'associazione tra gli agenti appartenenti ad un determinato gruppo e la vendita.

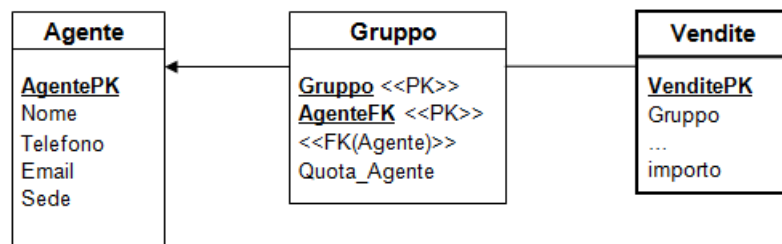


Figura 3.14: Esempio di utilizzo della tabella gruppo di agenti

L'associazione è mantenuta tramite l'utilizzo della chiave esterna per l'agente e la chiave primaria identificativa del gruppo. Si necessita inoltre dell'utilizzo di un attributo che specifichi la quota (cioè il contributo) che l'agente ha avuto sulla vendita associata ad un certo gruppo. Tipicamente si utilizzano valori compresi nell'intervallo $[0, 1]$ la cui somma, per uno stesso gruppo, è necessariamente 1. Utilizzando questa tecnica si deve però fare attenzione in fase di analisi, quando si definiscono le query SQL per l'estrazione dei risul-

tati. In alcuni casi, infatti, a seconda dell'analisi da svolgere, si deve tenere conto anche dell'attributo che specifica la quota dell'agente di un certo gruppo su una determinata vendita (ad esempio se si vuole conoscere l'importo totale della quota di vendita a cui ha contribuito un determinato agente). In altri casi invece, simile al precedente, l'attributo non deve essere considerato nell'interrogazione (ad esempio se si vuole conoscere l'importo totale delle vendite a cui un agente ha partecipato). Questa soluzione, utilizzabile per le analisi con lo strumento Pentaho C.E., pone però una maggior attenzione da parte degli utenti che definiscono le interrogazioni di analisi dei dati, evitando la generazione di risultati errati per motivi di interpretazione della richiesta.

3.8 La tabella per l'analisi del punto di pareggio

Quest'ultima Sezione del Capitolo è interamente dedicata alla progettazione della tabella utilizzata per il mantenimento dei dati utili alla determinazione del punto di pareggio tra costi e ricavi totali.

Trattato ampiamente in precedenza il tema e quindi definite le informazioni necessarie al raggiungimento dell'obiettivo, il passo successivo riguarda la modalità di acquisizione di tali dati.⁷

A tale proposito si è potuto scegliere tra due strade differenti:

1. una prima soluzione prevede la possibilità di lasciare direttamente allo strumento di reportistica la fase di acquisizione e calcolo dei dati necessari dalle tabelle dei fatti interessate e la conseguente determinazione del punto di pareggio. Sarà poi lo strumento stesso a generare e visualizzare correttamente il grafico desiderato per l'utente finale, una volta definite le sue caratteristiche.
2. la soluzione alternativa lascia allo strumento utilizzato per la generazione dei report la sola lettura dei dati già correttamente calcolati e la conseguente visualizzazione del grafico desiderato per l'utente.

Utilizzando la prima delle due alternative, la problematica maggiore da affrontare sta nell'elevato tempo di risposta del sistema. Ciò è dovuto soprat-

⁷Discusso nel Capitolo 2 alla Sezione 2.3.

tutto al fatto che lo strumento che genera il report viene utilizzato sia per il recupero dei dati dalle tabelle dei fatti tramite interrogazioni SQL, sia per la trasformazione di questi ultimi e l'applicazione dell'algoritmo di forecast.

Pertanto la scelta è ricaduta sulla seconda alternativa. In questo modo si sono ridotti sensibilmente i tempi di generazione del report, in quanto i dati visualizzati derivano da una interrogazione SQL su una tabella appositamente definita, mostrata in Figura 3.15, alimentata da flussi ETL precedentemente attivati.⁸ In altri termini, ogni volta che l'utente desidera visualizzare il report di analisi del punto di pareggio, dovrà solo attendere che il sistema legga le informazioni contenute nella tabella e che queste vengano correttamente associate al grafico corrispondente perchè i dati necessari sono già precalcolati e aggiornati ad ogni nuovo caricamento del data warehouse.

BreakEvenPoint
<u>ID Sede</u>
<u>Anno</u>
<u>Giorno</u>
CostiFissi
CostiVariabili
CostiTotali
RicaviTotali

Figura 3.15: Tabella per la break even analysis

Rimandando al Capitolo successivo le modalità di popolamento della tabella break even point, si definiscono brevemente i suoi attributi:

Un esempio delle informazioni contenute in tabella è riportato in Figura 3.7. Essa è costituita da 365 record identificabili per l'anno di riferimento (366 se l'anno è bisestile), il numero del giorno dell'anno riferito e la chiave univoca che identifica una singola sede.

Gli attributi rimanenti riportano l'importo dei costi che fanno parte della tipologia *costi fissi* sia diretti che indiretti. Il valore giornaliero è lo stesso riportato sulla totalità dei record in tabella; questo permetterà di poter disegnare sul piano cartesiano la retta costante parallela all'asse delle ascisse,

⁸Maggiori approfondimenti nel Capitolo 4 alla Sezione 4.2.

Nome	Descrizione	Tipo
ID_Sede	Valore identificativo della sede di riferimento	Intero
Anno	Valore dell'anno di riferimento	Intero
Giorno	Numero che identifica il giorno dell'anno, compreso nell'intervallo [1 - 365]	Intero
Costi Fissi	Valore costante giornaliero calcolato con l'algoritmo di forecast relativo alle voci di costo comprese nella tipologia costi fissi	Reale
Costi Variabili	Valore incrementale delle voci di costo comprese nella tipologia costi variabili	Reale
Costi Totali	Valore ottenuto dalla somma per riga dei costi fissi e costi variabili	Reale
Ricavi Totali	Valore incrementale delle voci di ricavo	Reale

Tabella 3.7: Descrizione attributi della tabella BreakEvenPoint

come mostrato in Figura 2.6. Al contrario l'importo dei *costi variabili* ha un andamento incrementale rispetto all'aumentare dei giorni dell'anno. In questo caso il calcolo è eseguito andando prima a definire l'importo unitario giornaliero sull'ammontare totale dei costi variabili e poi redistribuendo sui record tale importo moltiplicato per il numero del giorno corrispondente. Utilizzando questa tecnica è possibile definire la retta del piano corrispondente ai costi variabili come da Figura 2.6. La somma dei due importi appena descritti viene mantenuta nell'attributo *costi totali*. Essendo la grandezza dei costi variabili di tipo crescente, anche i costi totali mantengono la stessa caratteristica. In ultimo i *ricavi totali* rappresentano l'importo complessivo associato ai ricavi di vendita dei prodotti o dei servizi prestati. Anche in questo caso, trattandosi di una grandezza crescente, il valore unitario per il singolo record verrà calcolato con le stesse modalità utilizzate per i costi variabili.

A questo punto, una volta che l'utente ha scelto per quale sede (ad esempio Milano) e per quale anno (ad esempio 2011) vuole effettuare l'analisi del punto di pareggio, il sistema eseguirà una interrogazione SQL impostando sull'operatore FROM la tabella del punto di pareggio come parametro, e sull'operatore WHERE i valori di sede e anno selezionati. Dall'interrogazione verranno restituiti i 365 record corrispondenti comprensivi di tutti i valori

degli attributi appena discussi. Ciascun record può essere immaginato come un singolo punto appartenente all'asse delle ascisse sul piano cartesiano, a sua volta contenente quattro distinti valori riferiti all'asse delle ordinate. Proprio per come è stata costruita la tabella, una volta segnati sul piano tutti i 365 punti (record), automaticamente vengono rappresentate tutte e quattro le rette desiderate. Il punto di incrocio, se esiste, tra la retta dei costi totali e quella dei ricavi totali, traslato sull'asse delle ascisse, rappresenta il giorno dell'anno in cui si prevede il raggiungimento del punto di pareggio per una determinata sede in un certo anno.

Posticipando la discussione sulle formule matematiche e sugli algoritmi utilizzati nel prossimo Capitolo, ci limitiamo ora a stabilire che la tabella di Figura 4.13 mantiene le informazioni monetarie utili estratte da tutti e due i data mart definiti. La metodologia e le regole secondo cui gli importi a consuntivo e a budget si vanno a fondere per ottenere il risultato finale sono stati precedentemente affrontati nella presentazione dell'algoritmo di previsione.

3.9 Riassunto

- Il Capitolo è interamente dedicato alla progettazione del data warehouse per il budget d'azienda e si struttura secondo le fasi di progettazione discusse in [Albano 09a].
- Si mostra il caso di studio di interesse a partire dalla definizione del problema e analisi dei processi. Si discutono le esigenze e le richieste di interesse per il committente.
- A partire dalla raccolta dei requisiti di analisi si giunge alla progettazione concettuale dei data mart utilizzando, per la rappresentazione, il Dimensional Fact Model.
- Una volta definiti gli schemi concettuali si passa alla progettazione degli schemi logici, caratterizzati dalla tipica rappresentazione a stella. Sono riportate alcune soluzioni potenziali che si adattano maggiormente al caso di studio in oggetto e per ciascuna di esse si analizzano le caratteristiche, i pregi ed i difetti in previsione del loro utilizzo ai fini delle analisi dei dati.

- Scelta la soluzione più adeguata si mostra lo schema a costellazione del data warehouse che ne consegue.
- In relazione alla trattazione del punto di pareggio mostrato nel Capitolo precedente si presentano due possibili soluzioni alla problematica. La soluzione scelta prevede la progettazione di una tabella dedicata per il mantenimento dei dati di interesse e se ne mostra la struttura e le caratteristiche principali.

Capitolo 4

I flussi ETL

In questo Capitolo vengono presentate le fasi inerenti l'estrazione, la trasformazione e il caricamento dei dati nella struttura finale del data warehouse di Figura 3.7. Le Sezioni che seguono sono dedicate sia alla presentazione dello strumento utilizzato allo scopo, sia alla descrizione dei flussi stessi, distinguendo in particolar modo quelli utili al popolamento delle tabelle dei fatti e delle dimensioni del data warehouse, da quelli utili al popolamento della tabella per l'analisi del punto di pareggio.

4.1 Lo strumento Pentaho Data Integration

Il Pentaho Data Integration o Kettle, è lo strumento ETL fornito da Pentaho. Realizzato prettamente in Java e disponibile in un'unica distribuzione su più piattaforme (Linux, WinX, MAC OS), è costituito dalle seguenti componenti: *Spoon* (disegno grafico dei passi dell'ETL), *Pan* (esecuzione da linea di comando), *Kitchen* (scheduler) e *Carte* (esecuzione remota). La componente Spoon permette di realizzare graficamente le seguenti entità:

- Le *trasformazioni* che indicano come debbono essere raccolti, trasformati e caricati i dati. Sono composte da vari step connessi tra loro da collegamenti chiamati hop. Una tipica trasformazione inizia con uno step di input, uno o più step di trasformazione ed uno step di output.
- I *job* che sono simili alle trasformazioni poichè sono anch'essi composti da una serie di step collegati tra loro tramite degli hop, servono a

coordinare tra loro differenti trasformazioni ed attività. Un tipico job inizia con uno step di start e quindi una sequenza di step per lanciare script, trasformazioni ed eventualmente altri job annidati. I vari passi all'interno del job vengono eseguiti in sequenza.

All'interno della suite sono definiti già numerosi tipi di step, sia per le trasformazioni che per i job, organizzati per tipologia. Nell'evenienza in cui sia necessario definire nuovi step, questo è reso possibile utilizzando la programmazione in linguaggio JavaScript. Le trasformazioni ed i job realizzati vengono salvati dal software su una struttura chiamata *repository*, che può essere mantenuta su un data base o su una directory. Quando il repository è collocato su una directory questa contiene le trasformazioni e i job come singoli file XML, rispettivamente con il suffisso .ktr ed .kjb. In caso contrario, quando viene salvato su una base di dati relazionale, è composto da una serie di tabelle contenenti le informazioni (ad esempio r_transformation contiene tutte le trasformazioni). Il vantaggio dell'utilizzo di una base dati come repository è che tutte le trasformazioni ed i job possono essere condivisi da più utenti finali. Pan e Kitchen sono le due componenti che permettono di eseguire rispettivamente le trasformazioni e i job creati con Spoon. Infine Carte rappresenta un servizio web, che una volta attivato permette di eseguire in remoto i job.

Nel complesso, si può riassumere che un flusso ETL di Kettle corrisponde ad un file XML che a sua volta è prodotto tramite l'utilizzo dell'editor grafico Spoon messo a disposizione per l'utente. L'esempio di Figura 4.1 mostra le caratteristiche di tale interfaccia durante lo sviluppo di una trasformazione. Il riquadro rosso evidenzia l'area di sviluppo vera e propria del flusso mentre nel riquadro verde si mette in evidenza l'albero della trasformazione in oggetto. Come specificato dagli elementi contenuti nell'albero, per definire il flusso ETL per la trasformazione dell'esempio, occorre come primo passo creare le connessioni alle basi di dati (nel caso specifico l'unica connessione denominata *conn_dw_budget*). Una volta che la connessione è stata testata con successo si passa alla fase successiva di inserimento dei singoli passi utili nella trasformazione. L'interfaccia mette a disposizione un vasto elenco di passi distinti per categoria, rintracciabili nell'elenco degli oggetti di base evidenziato col colore blu nella figura. Tramite un semplice trascinamento

dell'oggetto all'interno dell'area di lavoro evidenziata in rosso, il passo va a far parte del flusso, naturalmente non prima di aver specificato tutte le caratteristiche ed i parametri di settaggio del passo stesso. Ogni volta che si inserisce un nuovo passo all'interno del flusso è essenziale che questo sia collegato sequenzialmente secondo il flusso di esecuzione tramite una connessione, detta hop, sia in ingresso che in uscita rispettivamente sul passo immediatamente precedente e sul passo immediatamente successivo. Terminato il flusso se ne può testare la corretta esecuzione tramite strumenti specifici di *debugging*, eventualmente riparando agli errori contenuti, e solo in seguito avviare le procedure di salvataggio della trasformazione (o del job) nel repository.

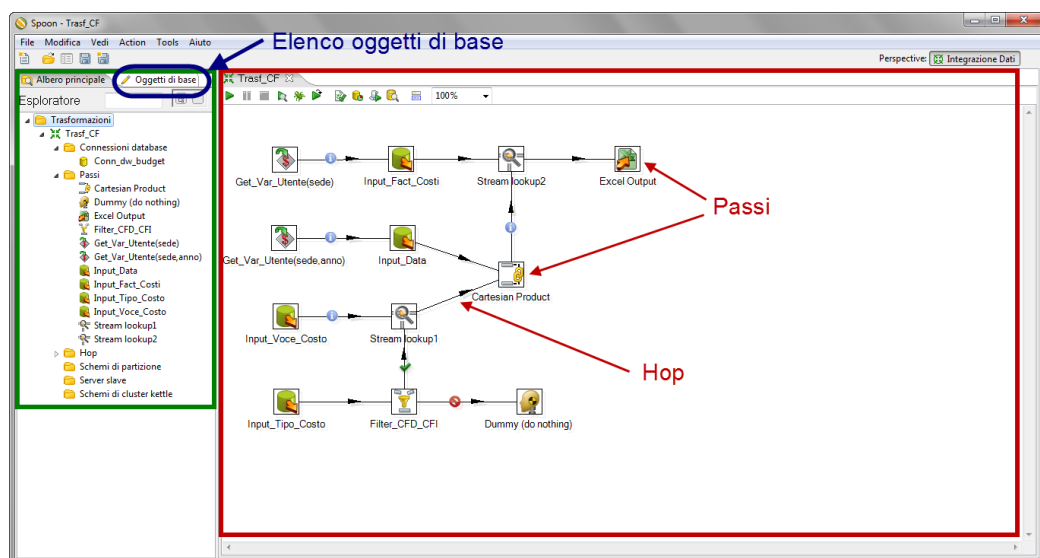


Figura 4.1: Esempio di trasformazione in Kettle

Tra le varie categorie di oggetti si trovano quelle dedicate alle fonti di input, all'output desiderato, alle modalità di unione tra due o più tabelle ecc. Nello specifico, tra gli elementi ai quali si fa ricorso molto frequentemente nello sviluppo di un flusso Kettle si trovano quelli della categoria *Input*, componenti utili alla definizione dell'estrazione dei dati, come i risultati di query su database relazionali, fogli di calcolo Excel, file di testo o file XML e CSV. All'interno di un processo tali componenti rappresentano la fase iniziale di estrazione dei dati. La categoria *Output* raccoglie elementi che permettono di aggiornare una base dati di destinazione come un database relazionale oppure di caricare gli stessi su fogli di calcolo Excel, file di testo, file XML e

CSV. In questo caso, all'interno di un processo, tali componenti rappresentano la fase conclusiva di caricamento dei dati. Infine, nella classe *Transform* sono contenuti la maggioranza dei componenti forniti dallo strumento utili per la fase di trasformazione vera e propria dei dati. Un'altra classe, i cui elementi sono frequentemente utilizzati, è la *Joins* che fornisce soluzioni diverse per eseguire giunzioni tra tabelle differenti sfruttando l'utilizzo di una chiave di giunzione. L'elenco completo delle categorie utilizzabili è fornito in Figura 4.2, con specifica evidenza delle componenti della classe di giunzione (ad esempio l'operatore *merge join* crea un prodotto cartesiano tra tabelle oppure il *sorted merge* ne stabilisce anche uno specifico ordinamento).

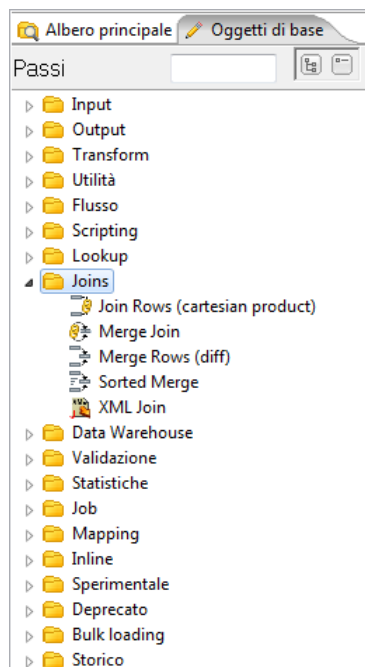


Figura 4.2: Passi base forniti dallo strumento

Nello stesso modo con cui vengono creati i flussi delle trasformazioni, si possono creare i job che, come osservato in precedenza, sono composti tipicamente da un'insieme di trasformazioni, singole operazioni o altri job. Il loro utilizzo è essenziale qualora si necessita l'aggiornamento di molteplici tabelle o la definizione di sequenze di trasformazioni. In riferimento a questo progetto, allo scopo di popolare o aggiornare lo schema a stella dei data mart di Figura 3.5 e 3.6 si definiscono singole trasformazioni per l'estrazione e il caricamento dei dati per ogni dimensione e tabella dei fatti. Ciascun gruppo di trasformazio-

ni viene poi raccolto all'interno di un job, detto job interno (come mostrato nell'esempio di Figura 4.3). A sua volta, al fine di definire una corretta sequenza delle trasformazioni, i vari job interni vengono incapsulati all'interno di un job esterno, detto job padre, identificabile dall'elemento di inizio della sequenza, chiamato *start*, e dall'elemento di fine sequenza, chiamato *successo*. Si nota, sempre in Figura 4.3, come ciascun elemento, job interno o singola trasformazione contenuta nel job padre, faccia parte di un'unica sequenza strettamente lineare di passi, questo perchè, al contrario delle trasformazioni, un job non permette l'avvio simultaneo di molteplici flussi di dati.

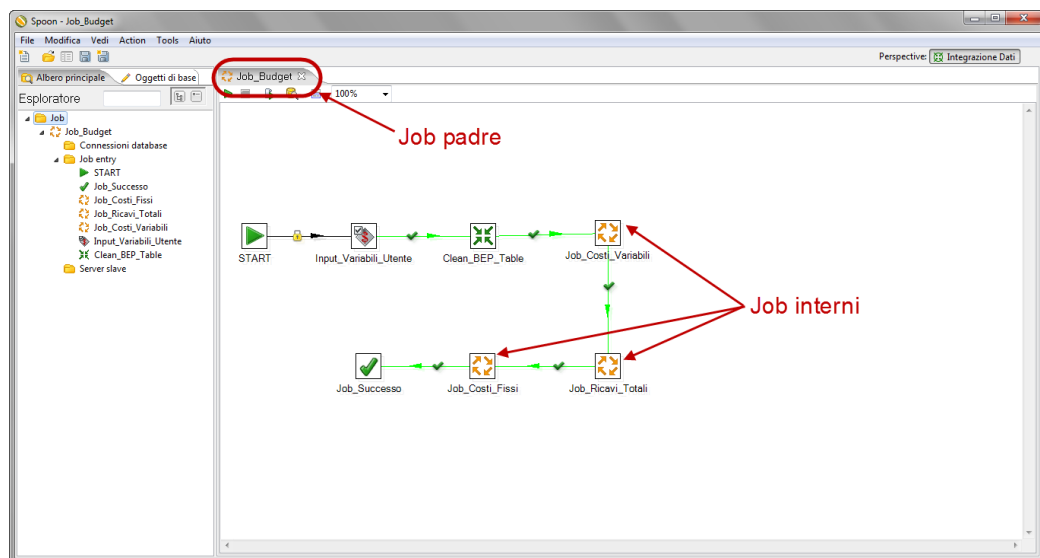


Figura 4.3: Esempio di job in Kettle

4.2 Estrazione, trasformazione e caricamento dei dati

Con l'acronimo ETL si intendono tutte quelle operazioni atte ad *estrarre* i dati dai sistemi sorgenti: database transazionali (OLTP), file di testo o qualsiasi altra fonte contenente i dati (ad esempio ERP o CRM). Una volta recuperati si passa alla *trasformazione*: pulendoli, integrandoli e omogeneizzandoli cosicché, una volta terminata l'elaborazione, essi possano essere *caricati* nella

struttura finale. Per quel che concerne questa tesi, possiamo distinguere due fasi di ETL principali:

- La prima riguardante il recupero, l'elaborazione e il caricamento nella struttura del data warehouse dei dati a partire dalle sorgenti esterne.
- La seconda rappresentante il popolamento della singola tabella per l'analisi del punto di pareggio a partire dalle informazioni presenti nel data warehouse precedentemente inizializzato.

Nei paragrafi successivi andremo ad analizzare l'implementazione dei due processi appena menzionati, concentrandoci particolarmente sul caricamento della tabella *BreakEvenPoint* in quanto ritenuto di maggiore interesse perché contenente anche aspetti economici oltre che lo studio e progettazione di un algoritmo di forecast. Tutto ciò è stato reso possibile attraverso lo strumento software Kettle messo a disposizione dalla suite di Pentaho e precedentemente descritto.

4.2.1 ETL: data warehouse

Come anticipato nel Capitolo precedente i dati utili al caricamento del data warehouse provengono da fonti eterogenee forniteci dal committente secondo le seguenti modalità:

- Valori a budget: sia i costi che i ricavi sono memorizzati e accessibili all'interno della base di dati del sistema gestionale usato dall'azienda.
- Valori a consuntivo: in questa circostanza è necessario fare una ulteriore distinzione:
 - I costi con cadenza mensile vengono prelevati dalla base di dati del sistema gestionale in uso dal committente.
 - I ricavi vengono estratti dalla base di dati del sistema CRM operazionale in dote al cliente, andando a recuperare gli importi delle opportunities terminate dagli agenti con successo (identificabili dal valore *closed won* dell'attributo *sales_stage*).¹

¹Per maggiori dettagli a riguardo del sistema CRM utilizzato dall'azienda cliente si rimanda alla lettura della Sezione 7.2 - "SugarCRM - Il Customer Relationship Management".

Vista l'impossibilità, per motivi di privacy, di poter usufruire delle reali informazioni contenute nella base di dati dei distinti sistemi aziendali del nostro cliente, ci sono stati forniti dei dati di demo che ricalcano gli originali. Queste informazioni sono presenti all'interno di una base di dati strutturata in modo analogo a quelle aziendali.

Mentre per i ricavi aziendali l'unica fonte deriva dai proventi delle vendite concluse positivamente dalla forza vendita, per i costi si è dovuto ricorrere ad una ulteriore distinzione. Infatti, l'azienda committente partiziona le voci di costo a seconda della loro tipologia in: costi diretti connessi all'attività degli agenti, costi generali riferiti alla gestione delle sedi e costi associabili ad altro personale. Quest'ultimi risultano essere trasversali alle sedi in quanto vengono definiti a livello globale. Essendo oggetto di studio di questo progetto l'analisi dei costi e dei ricavi in funzione degli agenti di vendita, si è reso indispensabile concordare con il committente una metodologia di ripartizione dei costi non direttamente imputabili ad essi. La scelta è così ricaduta sulla distinzione in:

- Costi generali: sono i costi non direttamente imputabili agli agenti ma riferiti ad una sede. Si tratta delle voci di costo quali ad esempio i canoni di locazione, le forniture di energia, le spese generali ecc. Per trattare queste informazioni si è scelto di suddividere equamente l'importo del costo generale associato ad una sede su tutti gli agenti ad essa associati.
- Costi dell'altro personale: sono i costi associati al personale dipendente non rientrante sotto la categoria agente. Questa categoria di costi non è imputabile nè agli agenti di vendita nè alle sedi in quanto definiti ad un livello più alto e quindi trasversali a quest'ultime. In questa circostanza si è scelto di creare un nuovo agente fittizio, identificabile tramite un codice univoco, associato ad una nuova sede anch'essa fittizia e anch'essa identificabile tramite un codice univoco. Questo modo di trattare le informazioni interessate implica però un'attenzione particolare in sede di analisi dei dati e generazione di report.

Fatte le dovute assunzioni si possono ora osservare i flussi ETL definiti, distinguendo le trasformazioni utilizzate per il popolamento delle tabelle dimensionali da quelle utilizzate per il caricamento delle due tabelle dei fatti.

In Figura 4.4 è rappresentato lo schema del flusso principale che raccoglie tutte le trasformazioni necessarie.

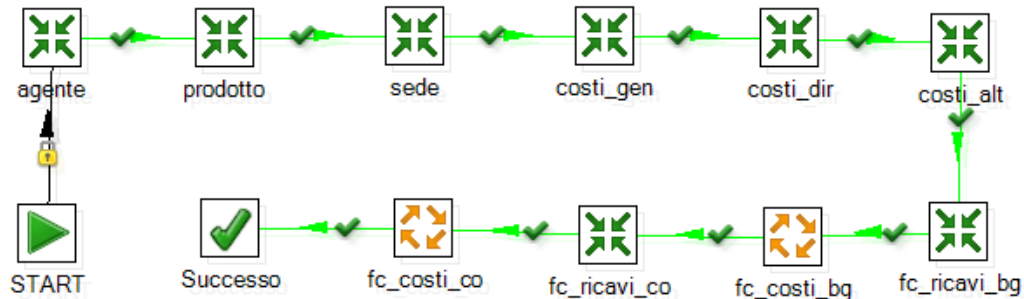


Figura 4.4: Job esterno per il caricamento del data warehouse

Si nota come nella prima parte del flusso le trasformazioni riguardano il caricamento delle tabelle dimensionali, nell'ordine: agente, prodotto, sede, costi generali, costi diretti e costi altro personale. Ciascuna di queste trasformazioni comporta la lettura dei dati dalle rispettive tabelle della base di dati e la selezione dei campi necessari; in ultimo l'inserimento nel data warehouse.

Successivamente si sono generati i flussi per il caricamento delle due tabelle dei fatti: costi e ricavi, sia per quanto riguarda i valori a budget che a consuntivo. Per i costi, come precedentemente discusso, si è dovuto procedere distinguendo tre differenti trasformazioni in base alla tipologia.

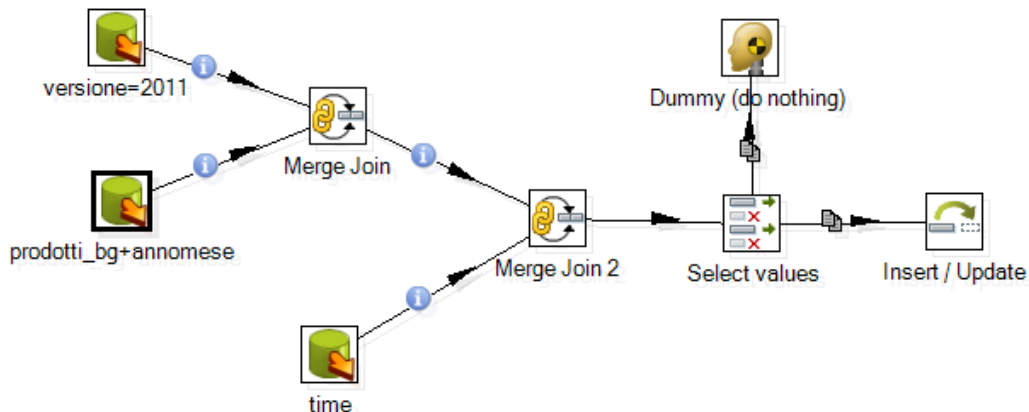


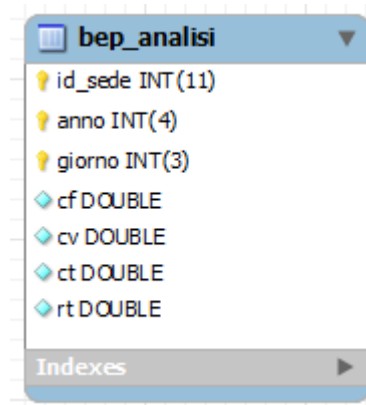
Figura 4.5: Caricamento dei dati a budget nella tabella dei fatti ricavi

Per quanto riguarda i ricavi invece è stata sufficiente un'unica trasformazione, dal momento che queste informazioni godono di un minor numero di passi

di trasformazione e manipolazione delle informazioni in quanto la situazione risulta meno articolata rispetto al caso precedente. A tale proposito in Figura 4.5 si riporta a titolo esemplificativo il flusso creato per i ricavi a budget, essendo molto simile a tutti gli altri.

4.2.2 ETL: punto di pareggio

Partendo dalla struttura del data warehouse caricata, come illustrato nella precedente Sezione, si tratta adesso di ricavarne le informazioni necessarie per effettuare l'analisi del punto di pareggio. Al termine di questo flusso ETL l'obiettivo è quello di creare e popolare la tabella per l'analisi del punto di pareggio (denominata BreakEvenPoint) la cui struttura è mostrata in Figura 4.6.²



The image shows a screenshot of a database table structure for a table named 'bep_analisi'. The table has the following columns:

Column Name	Data Type
id_sede	INT (11)
anno	INT (4)
giorno	INT (3)
cf	DOUBLE
cv	DOUBLE
ct	DOUBLE
rt	DOUBLE

At the bottom of the screenshot, there is a section labeled 'Indexes' with a right-pointing arrow, indicating that the index structure is not fully visible.

Figura 4.6: Struttura della tabella BreakEvenPoint

Lo scopo finale è individuare il giorno in cui si verificherà, se esiste, il punto di pareggio. Inizialmente verranno sfruttati solo i dati presenti a budget, ma mano a mano che durante l'anno si rendono disponibili i valori a consuntivo si ricalcherà il nuovo punto di equilibrio fra costi totali e ricavi, ponderando i dati a consuntivo con quelli budgetari attraverso l'algoritmo presentato nella Sezione 2.3. Al termine saremo quindi in grado di rappresentare su un piano cartesiano, per ciascuna sede, il proprio punto di pareggio per un determinato anno. Risulta quindi evidente che *anno* e *id_sede* sono due parametri che il

²Per la quale valgono le considerazioni fatte nel Capitolo precedente Sezione 3.8.

processo ETL si aspetta in input per poter iniziare l'elaborazione (Figura 4.7).



Figura 4.7: Step di impostazione variabili di input (anno, id_sede)

Il primo job realizzato in kettle, quello più esterno, contiene tutte le trasformazioni e gli altri job del flusso e prende il nome di *job_bep*, rappresentato nella Figura 4.8.



Figura 4.8: Rappresentazione grafica del flusso Job padre

Il nodo *canc_bep* ha una duplice funzione:

- Inizializza una variabile globale con il numero di giorni dell'anno passato come parametro, tenendo conto se si tratta di un anno bisestile.
- Controlla se nella tabella denominata BreakEvenPoint esistono già delle righe associate a quella sede per quell'anno. In caso affermativo vengono cancellate, ciò significa che il flusso era già stato lanciato in precedenza per il calcolo del punto di pareggio. In caso negativo non vengono effettuate cancellazioni e lo step non esegue nessuna azione.



Figura 4.9: Job dei costi

Il job successivo, *job_COSTI*, contiene le tre trasformazioni mostrate in Figura 4.9, che permettono a partire dai dati all'interno del data warehouse di calcolare rispettivamente:

- *costi_co*: il valore giornaliero a consuntivo dei costi fissi e variabili. Sommando prima i costi fissi a consuntivo e dividendoli per il massimo giorno dell'anno in cui è stato sostenuto il costo e ripetendo analoga formula per i costi variabili, otteniamo i due valori medi giornalieri, rispettivamente dei costi fissi e variabili.

$$co_cf = \frac{\sum cf_consuntivo}{maxGiorno_co} \quad co_cv = \frac{\sum cv_consuntivo}{maxGiorno_co}$$

- *costi_bg*: il valore giornaliero a budget dei costi fissi e variabili. Analogamente a sopra si calcolano i due valori di budget giornalieri medi riferiti ai costi fissi e variabili.

$$bg_cf = \frac{\sum cf_budget}{GiorniAnno} \quad bg_cv = \frac{\sum cv_budget}{GiorniAnno}$$

- *forecast_c*: il valore giornaliero di forecast dei costi fissi e variabili. Partendo dai valori sopra calcolati ed applicando la formula definita nel Capitolo 2 otteniamo sia per i costi fissi che per i costi variabili un valore che tiene in considerazione gli importi a consuntivo e a budget ponderati in funzione del loro numero di giorni.

$$\frac{forecast_cf_gg = (co_cf * maxGiorno_co) + (bg_cf * (GiorniAnno - maxGiorno_co))}{GiorniAnno}$$

$$\frac{forecast_cv_gg = (co_cv * maxGiorno_co) + (bg_cv * (GiorniAnno - maxGiorno_co))}{GiorniAnno}$$

Al termine del job_COSTI verrà restituita come output una t-upla (nome, valore) così composta: (forecast_cf_gg, valore; forecast_cv_gg, valore).

Analogamente la procedura viene ripetuta per i ricavi, all'interno del job successivo *job_RT* (il quarto nodo rappresentato in Figura 4.8). In questo caso però, non essendo presente la distinzione tra ricavi fissi e variabili, la procedura all'interno delle tre trasformazioni, mostrate in Figura 4.10, risulta più snella.

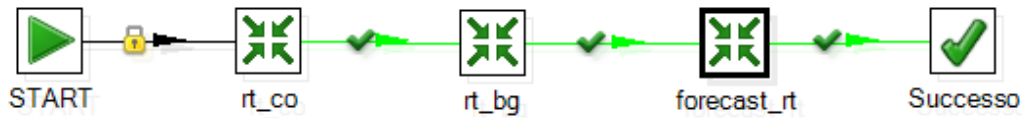


Figura 4.10: Job dei ricavi

L'output restituito dal flusso `job_RT` è una t-upla formata dal valore di previsione giornaliero, del tipo: (`forecast_rt_gg`, valore).

Entrando sempre più nello specifico, si analizza la composizione della trasformazione `rt_co` di Figura 4.10, mostrata in Figura 4.11.

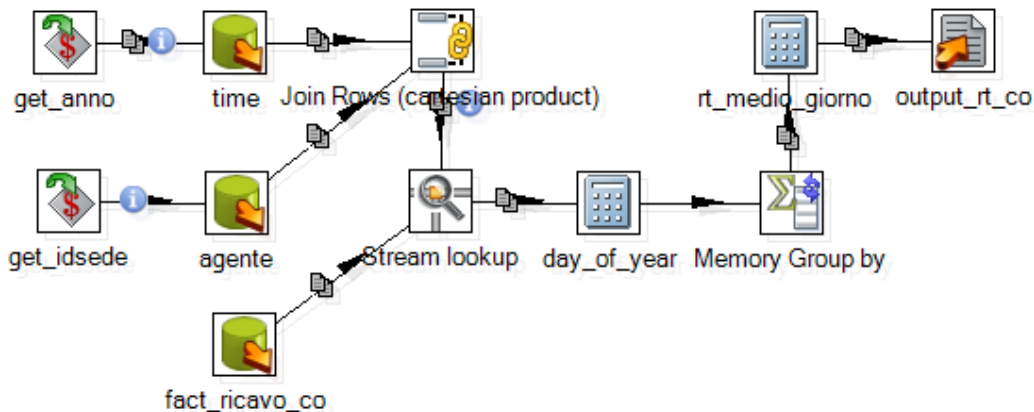


Figura 4.11: Trasformazione per il computo del valore giornaliero dei ricavi

Questa è utilizzata per il calcolo del valore giornaliero a consuntivo dei ricavi, tenendo in considerazione che anche la trasformazione successiva, per il computo dei valori dei ricavi a budget, segue una struttura conforme.

Il flusso comincia con i nodi `get_anno` e `get_idsede`, la cui funzione è quella di recuperare dal job padre (`job_bep`) i parametri precedentemente definiti riferiti all'anno e alla sede. Il passo successivo prevede la lettura delle righe interessate dalle tabelle `time` e `agente` del data warehouse, filtrate rispettivamente per i parametri `anno` e `idsede` appena acquisiti (l'oggetto che implementa questa funzionalità sfrutta l'esecuzione di query tramite una particolare sintassi mostrata in Figura 4.12).

In seguito si esegue il prodotto cartesiano fra gli output sopra descritti e successivamente una giunzione con la tabella `fact_ricavo_co` sui campi chiave `id_agente` e `id_sede`. Da questa sequenza di step si ottengono tutti gli importi

dei ricavi a consuntivo per gli agenti associati a una specifica sede in un determinato anno.

SQL	SQL
<pre>SELECT id_time , date FROM time WHERE anno=?</pre>	<pre>SELECT id_agente , id_sede FROM agente WHERE id_sede=?</pre>

Figura 4.12: Esempio di interrogazioni SQL

Si procede poi raggruppando sulla chiave *id_sede* e applicando la funzione di aggregazione *sum* e *max* rispettivamente sull'attributo *importo* e *data*. Il flusso termina eseguendo il rapporto tra $sum(importo)$ e $max(data)$, andando ad ottenere come output finale l'importo del ricavo a consuntivo giornaliero medio.

$$co_{rt} = \frac{\sum rt_{consuntivo}}{maxGiorno_{co}}$$

Una volta ottenute le due t-uple, quella del forecast dei costi (*forecast_cf_gg*, valore; *forecast_cv_gg*, valore) e quella del forecast dei ricavi (*forecast_rt_gg*, valore); i rispettivi valori possono essere scritti nella tabella *BreakEvenPoint* tramite un'operazione di *insert*, che avviene nell'ultimo job della Figura 4.8.

Quest'ultima operazione viene effettuata tenendo in considerazione quanto segue:

- Il valore del forecast dei costi fissi, essendo appunto fisso e quindi costante durante tutto l'arco dell'anno, viene ripetuto per tutti i giorni (colonna *cf* in Figura 4.13).
- Il valore del forecast dei costi variabili e dei ricavi viene inserito incrementalmente nella tabella, seguendo questi criteri:

$$valore_inserito_cv = forecast_cv_gg * giorno_anno$$

$$valore_inserito_rt = forecast_rt_gg * giorno_anno$$

l'applicazione di queste formule assicurano che il valore di forecast sia direttamente proporzionale al numero del giorno dell'anno cui è riferito (come visibile rispettivamente nelle colonne *cv* e *rt* della Figura 4.13).

- Il valore del forecast del costo totale è invece un campo calcolato definito come somma del costo fisso e del costo variabile relativo a quel giorno.

id_sede	anno	giorno	cf	cv	ct	rt	date
1	2011	1	266640.76712	893.33109	267534.09821	2068.25371	2011-01-01
1	2011	2	266640.76712	1786.66218	268427.4293	4136.50742	2011-01-02
1	2011	3	266640.76712	2679.99327	269320.76039	6204.76113	2011-01-03
1	2011	4	266640.76712	3573.32436	270214.09148	8273.01484	2011-01-04
1	2011	5	266640.76712	4466.65545	271107.42257	10341.26855	2011-01-05
1	2011	6	266640.76712	5359.98654	272000.75366	12409.52226	2011-01-06
1	2011	7	266640.76712	6253.31763	272894.08475	14477.77597	2011-01-07
1	2011	8	266640.76712	7146.64872	273787.41584	16546.02968	2011-01-08
1	2011	9	266640.76712	8039.97981	274680.74693	18614.28339	2011-01-09
1	2011	10	266640.76712	8933.3109	275574.07802	20682.5371	2011-01-10
1	2011	11	266640.76712	9826.64199	276467.40911	22750.79081	2011-01-11
1	2011	12	266640.76712	10719.97308	277360.7402	24819.04452	2011-01-12
1	2011	13	266640.76712	11613.30417	278254.07129	26887.29823	2011-01-13
1	2011	14	266640.76712	12506.63526	279147.40238	28955.55194	2011-01-14
1	2011	15	266640.76712	13399.96635	280040.73347	31023.80565	2011-01-15
1	2011	16	266640.76712	14293.29744	280934.06456	33092.05936	2011-01-16
1	2011	17	266640.76712	15186.62853	281827.39565	35160.31307	2011-01-17
1	2011	18	266640.76712	16079.95962	282720.72674	37228.56678	2011-01-18
1	2011	19	266640.76712	16973.29071	283614.05783	39296.82049	2011-01-19
1	2011	20	266640.76712	17866.6218	284507.38892	41365.0742	2011-01-20
1	2011	21	266640.76712	18759.95289	285400.72001	43433.32791	2011-01-21
1	2011	22	266640.76712	19653.28398	286294.0511	45501.58162	2011-01-22
1	2011	23	266640.76712	20546.61507	287187.38219	47569.83533	2011-01-23
1	2011	24	266640.76712	21439.94616	288080.71328	49638.08904	2011-01-24
1	2011	25	266640.76712	22333.27725	288974.04437	51706.34275	2011-01-25

Figura 4.13: Porzione della tabella BreakEvenPoint popolata

In Figura 4.13 è rappresentata una porzione della tabella denominata BreakEvenPoint popolata dal flusso ETL appena descritto.

Ultimata la fase di inserimento dati nella tabella BreakEvenPoint termina anche la seconda fase di ETL relativa al nostro progetto e si può passare

alla presentazione della fase successiva. Nei prossimi Capitoli si tratta dettagliatamente il processo vero e proprio di analisi dei dati che popolano il data warehouse e la conseguente generazione di documenti di reportistica.

4.3 Riassunto

- Il Capitolo tratta il processo di estrazione, trasformazione e caricamento dei dati nel data warehouse. Viene presentato lo strumento della Suite Pentaho C.E. utilizzato allo scopo.
- Si mostrano gli schemi dei flussi di dati progettati a partire dalle fonti eterogenee di informazioni dalle quali vengono estratti, fino al loro caricamento nel data warehouse.
- Per comodità prettamente progettuali, i flussi sono distinti in due macro categorie. Il primo tratta esclusivamente i dati a budget mentre il secondo quelli a consuntivo, in entrambe le circostanze distinguendo le informazioni di costo da quelle di ricavo.
- Sfruttando le caratteristiche dello strumento di BI utilizzato, i flussi sono stati riorganizzati in base al tipo di destinazione, distinguendo i flussi che popolano le dimensioni del data warehouse da quelli che popolano le tabelle dei fatti.
- A livello utente è presentato un unico flusso principale (detto Job padre o esterno) che raccoglie in sequenza l'insieme dei flussi definiti.
- Per quanto riguarda la trattazione del punto di pareggio, sono mostrati i flussi che popolano la tabella di riferimento e l'insieme delle formule implementate all'interno delle trasformazioni dei dati al fine di ottenere le informazioni necessarie allo studio.

Capitolo 5

Le analisi dei dati

Una volta definita la struttura del data warehouse e popolata delle informazioni utili, il progetto prosegue con la fase di sviluppo dei modelli utilizzati durante l'analisi vera e propria dei dati. Le Sezioni seguenti sono interamente dedicate alla comprensione del sistema di analisi utilizzato e alla presentazione del modello multidimensionale a cubo sfruttato sia per la rappresentazione dei dati di costo che per quelli di ricavo.

5.1 Il sistema di analisi dei dati

Le analisi multidimensionali dei dati costituiscono un tema di largo interesse nel campo della business intelligence. Tramite opportuni strumenti interattivi si rende possibile l'estrazione di dati di sintesi che possono essere esplorati da molteplici punti di vista al fine di scoprire determinate situazioni anomale o tendenze interessanti nella gestione dei processi aziendali [Albano 09a].

Al fine di comprendere questo importante concetto di multidimensionalità dell'informazione è necessario fare distinzione tra applicazioni transazionali interattive che utilizzano basi di dati (in altri termini *OLTP*, *On Line Transaction Processing*) e applicazioni dedicate al supporto alle decisioni che utilizzano data warehouse (definite come *OLAP*, *On Line Analytical Processing*).

- **OLTP**

Acronimo di On Line Transaction Processing (elaborazione delle transazioni in linea). Rappresenta il principale uso nei sistemi *DBMS* per

la tradizionale elaborazione di transazioni che realizzano i processi operativi per il funzionamento di organizzazioni. Tipicamente sono caratterizzati da: operazioni predefinite e relativamente semplici, ogni operazione coinvolge poche quantità di dati e tali dati sono da considerarsi di dettaglio e aggiornati. Un classico schema è riportato in Figura 5.1 [Albano 05b]. Indica un database relazionale ideale per la gestione di dati in evoluzione. Generalmente le informazioni mantenute nella base di dati sono utilizzate da un numero elevato di utenti che eseguono contemporaneamente transazioni che comportano la modifica dei dati in tempo reale e tipicamente le singole richieste di dati da parte degli utenti fanno riferimento ad un basso numero di righe. Le basi di dati OLTP sono progettate per consentire alle applicazioni transazionali di scrivere solo i dati necessari per la gestione di una singola transazione nel modo più rapido possibile. In un contesto di gestione aziendale tipici sistemi sono rintracciabili ad esempio durante i processi di pagamento ai fornitori, registrazione delle fatture, emissione di ordini di acquisto, registrazione delle bolle, registrazione dei clienti ecc. I sistemi OLTP forniscono parte dei dati e delle informazioni utilizzate nei sistemi OLAP; costituiscono quindi una classica sorgente dati per quest'ultimi.

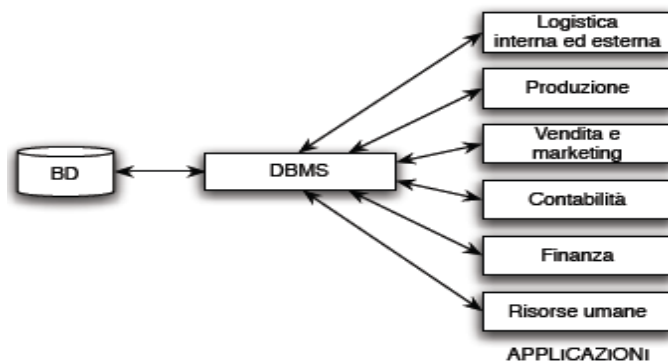


Figura 5.1: Esempio di sistema informatico operativo

- **OLAP**

Acronimo di On Line Analytical Processing. Rappresenta il principale uso nei sistemi data warehouse utile per le analisi dei dati di supporto alle decisioni. Tipicamente sono caratterizzati da operazioni complesse

e casuali che possono coinvolgere un gran numero di dati, spesso trattati sotto forma di aggregazioni e tipicamente rappresentanti sequenze storiche (i dati possono essere anche non attuali). Un classico schema è riportato in Figura 5.2 [Albano 05b].

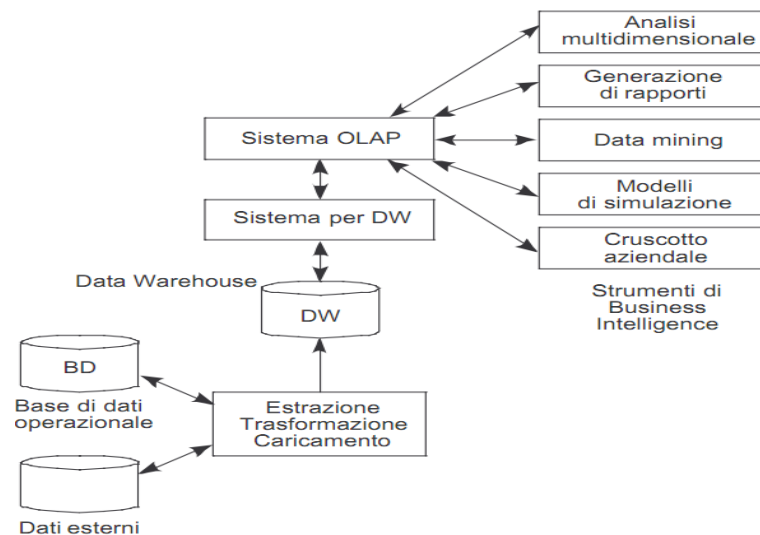


Figura 5.2: Esempio di sistema informatico direzionale

L'introduzione del termine OLAP si deve all'autore E.F.Codd nel rapporto "Providing OLAP to User Analysts: An IT Mandate" del 1993 in cui tale concetto è descritto in 12 regole: *vista concettuale multidimensionale, trasparenza, accessibilità, coerenza nelle performance dei rapporti, architettura client/server, dimensionalità generiche, manipolazione matrici sparse, supporto multi-utente, illimitate operazioni attraverso le dimensioni, manipolazione intuitiva dei dati, rapporti flessibili, illimitati livelli di aggregazione e dimensioni*.¹ Nel complesso, si può affermare che il termine OLAP viene utilizzato per rappresentare un nuovo approccio all'analisi dei dati di supporto alle decisioni per consentire ai dirigenti di passare dall'uso dei tradizionali e numerosi rapporti statici stampati periodicamente su carta, a rapporti in formato elettronico modificabili interattivamente per ottenere rapidamente risposte a nuove richieste di analisi dei dati [Albano 09a]. Solo in periodi

¹In questo contesto ci si limita al semplice elenco delle 12 regole che caratterizzano un sistema OLAP. Maggiori dettagli ed il rapporto per intero può essere consultato in rete.

successivi, l'autore N.Pendse, nel documento "The OLAP Report" del 2004 propone il modello FASMI, acronimo di *Fast Analysis of Shared Multidimensional Information*, in cui i sistemi OLAP sono descritti in 5 termini chiave: *veloce, analisi, condiviso, multidimensionale, informazione*.² In letteratura si distingue un *server OLAP* in una delle sue tre forme principali: MOLAP (*Multidimensional OLAP*) quando mantiene in memoria locale sia i dati del cubo che gli aggregati del cubo esteso sfruttando opportune strutture matriciali, ROLAP (*Relational OLAP*) quando sia i dati del cubo che i suoi valori aggregati sono mantenuti in un database relazionale, HOLAP (*Hybrid OLAP*) rappresenta una combinazione delle due tipologie precedenti e quindi mantiene in memoria locale i dati aggregati e lascia i dati del cubo su un database relazionale.

Per concludere, si riassumono alcune tra le più importanti differenze tra i due sistemi appena discussi in Tabella 5.1.

	OLTP	OLAP
Scopi	Supporto operatività	Supporto alle decisioni
Utenti	Molti, esecutivi (impiegati o tecnici)	Pochi (dirigenti o analisti)
Organizzazione	Per settori	Per soggetti
Usi	Predefiniti, noti a priori	Estemporanei, poco prevedibili
Dati	Attuali, dettagliati, relazionali, specifici	Storici, sintesi, multidimensionali, integrati
Tipi di accessi	Brevi transazioni con letture e scritture	Letture con ricerche complesse
Numero utenti	Alto	Basso
Quantità di dati per attività	Bassa, dell'ordine delle decine	Alta, dell'ordine di milioni
Ottimizzazioni	Per transazioni	Per analisi dei dati

Tabella 5.1: Principali differenze tra sistemi OLTP e OLAP

²Anche in questo caso ci si limita all'elenco delle caratteristiche FASMI. Il documento riferito può essere rintracciato e consultato in rete.

5.2 Il modello multidimensionale a cubo

Da quanto detto nella Sezione precedente, si evidenzia che il concetto di multidimensionalità del dato risulta essere un requisito fondamentale nei sistemi OLAP e tendenzialmente ci si riferisce a questa caratteristica dei dati da loro gestiti con il termine di cubo. In un modello multidimensionale a cubo si usa rappresentare un generico fatto del data warehouse ad n dimensioni come un punto all'interno di uno spazio n -dimensionale. In altri termini, un fatto rappresentabile come un punto dello spazio è individuabile dai valori delle dimensioni che lo caratterizzano e determina un insieme di valori rappresentanti le misure associate a quel fatto. Una tipica rappresentazione a cubo di un fatto è mostrata in Figura 5.3.

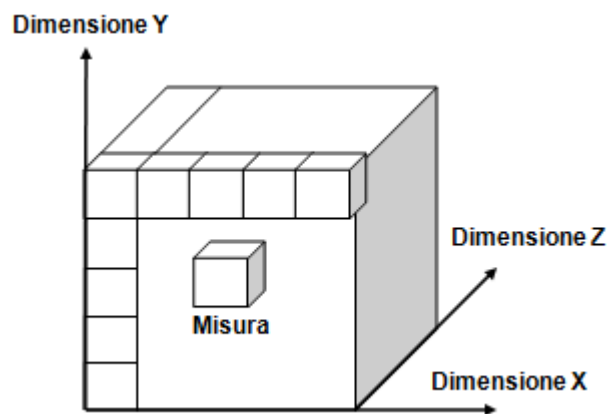


Figura 5.3: Esempio di cubo di dati a tre dimensioni e una misura

5.2.1 Gli strumenti della suite Pentaho C.E.

Per quanto riguarda la fase di progettazione dei cubi multidimensionali sui dati riferiti al progetto oggetto di questa tesi, la *Suite Pentaho C.E.* si avvale dell'utilizzo di tre strumenti:

1. *Mondrian* (motore ROLAP di Pentaho C.E.).
2. *Schema Workbench* (interfaccia utente di progettazione dei cubi).
3. *JPivot* (interfaccia utente per la visualizzazione dei risultati).
4. *Saiku* (interfaccia alternativa per la visualizzazione dei risultati).

Mondrian

Nel seguito si discutono le caratteristiche del motore ROLAP citato e le modalità di utilizzo di entrambe le interfacce, la prima utilizzata da progettisti e sviluppatori, la seconda utilizzata dagli utenti finali.

Sinteticamente si può affermare che Mondrian è un motore ROLAP scritto in *Java* (in quanto memorizza esclusivamente la struttura dei dati, quindi le loro correlazioni, ma lascia i dati persistenti nella base di dati) e progettato per eseguire interrogazioni scritte nel linguaggio *MDX*.³ Questo fa sì che la lettura dei dati possa essere effettuata su molteplici basi di dati RDBMS (tra cui anche la base di dati MySQL utilizzata in questo progetto) e la presentazione dei risultati in un formato multidimensionale avviene attraverso una *API Java*.

Il sistema Mondrian presenta quattro livelli: il *presentation layer*, il *dimensional layer*, lo *star layer* e infine lo *storage layer*, come rappresentato in Figura 5.4.

Il *presentation layer* determina ciò che l'utente finale vede sul proprio monitor e la modalità di interazione per effettuare nuove interrogazioni al sistema.

Il secondo strato è il *dimensional layer*. Ha lo scopo di analizzare, validare ed eseguire query MDX. Una componente ulteriore permette all'applicazione di poter manipolare le interrogazioni SQL già definite evitando di dover generare nuove istruzioni MDX per ogni nuova richiesta di dati.

Il terzo strato è lo *star layer*, responsabile del mantenimento in memoria di un *buffer* che contiene i dati riferite con più frequenza.

Per finire, lo strato di *storage layer* è un RDBMS.

Questi componenti possono coesistere sulla stessa macchina, oppure possono essere distribuiti tra macchine distinte. Questo non vale però per il

³Il tema è ampiamente trattato sulla pagina <http://mondrian.pentaho.com/> ed in questo contesto si riporta la seguente definizione: “MDX è l'acronimo di *Multi-Dimensional Expressions*, introdotto da Microsoft con Microsoft SQL Server OLAP Service intorno all'anno 1998 come componente del linguaggio OLE DB per API OLAP. Più recentemente MDX compare come parte di XML per Analysis API”. Un esempio della sintassi utilizzata nelle query MDX può essere il seguente: “SELECT [Misure].[Quantità vendute], [Misure].[Quantità rimanenti] ON COLUMNS, [Prodotto].nome ON ROWS FROM [Vendite] WHERE [Data].[anno].[trimestre]”.

dimensional layer e lo *star layer* perchè comprendendo il server Mondrian devono essere mantenuti sulla stessa macchina. Lo strato *storage layer* invece potrebbe risiedere su un'altra macchina e reso accessibile tramite una connessione remota *JDBC*.

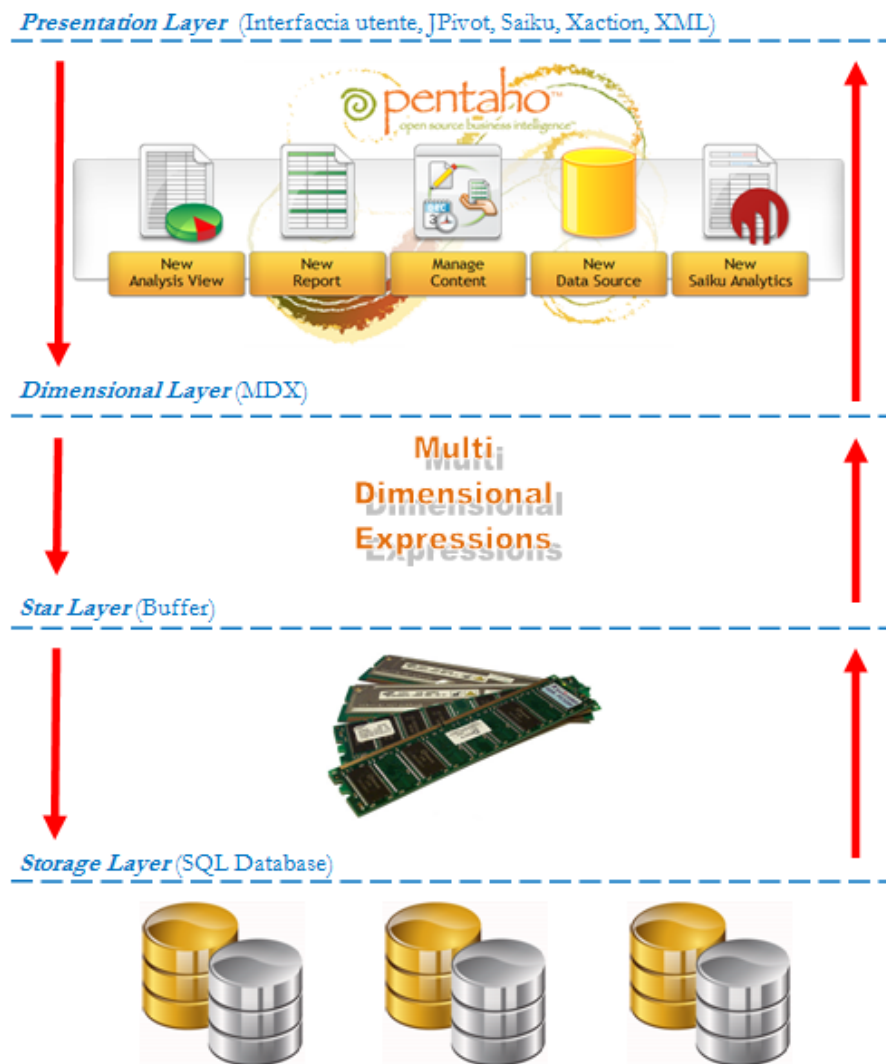


Figura 5.4: Schema architetturale del motore Mondrian ROLAP

Schema Workbench

Generalmente le analisi dei dati vengono effettuate dagli utenti sviluppatori sfruttando opportune interfacce grafiche in grado di formulare le richieste che, in un secondo momento, verranno automaticamente tradotte in interrogazio-

ni SQL sul data warehouse. A tale proposito, la suite Pentaho C.E. mette a disposizione l'interfaccia grafica Schema Workbench, utile alla definizione, creazione e test degli schemi Mondrian dei cubi. Uno schema, tecnicamente un file XML, è da considerarsi un cubo che sfrutta una o più tabelle dei fatti e una o più dimensioni presenti in un data warehouse. Nell'esempio di Figura 5.5 si mostra il file XML generato a seguito dell'utilizzo, da parte dell'utente progettista, dell'interfaccia Schema Workbench.

```

1. <Schema name="dw_budget">
2.   <Dimension type="StandardDimension" highCardinality="false" name="time">
3.     <Hierarchy name="data" hasAll="true" allMemberName="all" primaryKey="id_time">
4.       <Table name="time" />
5.       <Level name="anno" table="time" column="anno" type="String" uniqueMembers="true"
6.         levelType="Regular" hideMemberIf="Never" />
7.       <Level name="semestre" table="time" column="semestre" type="String" uniqueMembers="false"
8.         levelType="Regular" hideMemberIf="Never" />
9.       <Level name="trimestre" table="time" column="trimestre" type="String" uniqueMembers="false"
10.        levelType="Regular" hideMemberIf="Never" />
11.       <Level name="mese" table="time" column="nomemese" ordinalColumn="mese" type="String"
12.        uniqueMembers="false" levelType="Regular" hideMemberIf="Never" />
13.     </Hierarchy>
14.   </Dimension>

15. <Dimension type="StandardDimension" highCardinality="false" name="agente">
16.   <Hierarchy name="agente" hasAll="true" allMemberName="all" primaryKey="id_agente">
17.     <Table name="agente" />
18.     <Level name="cognome" table="agente" column="cognome" type="String" uniqueMembers="true"
19.       levelType="Regular" hideMemberIf="Never" />
20.   </Hierarchy>
21. </Dimension>

22. <Cube name="ricavo" cache="true" enabled="true">
23.   <Table name="fact_ricavo" />
24.   <DimensionUsage source="time" name="time" foreignKey="id_time" highCardinality="false" />
25.   <DimensionUsage source="agente" name="agente" foreignKey="id_agente"
26.     highCardinality="false"/>
27.   <Dimension type="StandardDimension" foreignKey="id_prodotto" highCardinality="false"
28.     name="prodotto">
29.     <Hierarchy name="prodotto" hasAll="true" allMemberName="all" primaryKey="id_prodotto">
30.       <Table name="prodotto" />
31.       <Level name="nome" table="prodotto" column="nome" type="String" uniqueMembers="true"
32.         levelType="Regular" hideMemberIf="Never" />
33.     </Hierarchy>
34.   </Dimension>
35.   <Measure name="budget_r" column="importo_bg" aggregator="sum" visible="true" />
36.   <Measure name="consuntivo_r" column="importo_co" aggregator="sum" visible="true" />
37.   <CalculatedMember name="scostamento_r" formula="Measures.[consuntivo_r]-
38.     Measures.[budget_r]" dimension="Measures" visible="true" />
39. </Cube>
40. </Schema>

```

Figura 5.5: Esempio di schema XML generato tramite Schema Workbench

Nel complesso questo file di output si compone di un'insieme di *tag* standard con i quali si vanno a definire le dimensioni e le misure che si vogliono rappresentare nel cubo. Qualora in uno stesso schema si necessita la definizione di più cubi, Schema Workbench provvederà semplicemente ad aggiungere un

nuovo tag `< Cube > ... < /Cube >` al cui interno sono dichiarate le nuove tabelle dei fatti, delle dimensioni e le misure. Se invece le analisi da effettuare richiedono il recupero di informazioni da molteplici tabelle dei fatti lo strumento offre la possibilità di definire una vista virtuale sui cubi all'interno dei tag `< VirtualCube > ... < /VirtualCube >`. Questa tecnica è però utilizzabile solo se le tabelle dei fatti coinvolte hanno in comune una o più tabelle delle dimensioni di analisi (nel caso considerato si osserva dagli schemi di Figura 3.5 e 3.6 la condivisione delle dimensioni *Data* e *Agente*). Per concludere le gerarchie dimensionali presenti nelle tabelle vengono dichiarate mediante l'utilizzo dei tag `< Hierarchy > ... < /Hierarchy >`, i cui livelli sono scanditi dal tag `< Level >`.

Come detto, il file XML presentato è l'output derivante dall'utilizzo dello strumento in dotazione al progettista fornito da Pentaho C.E, la cui interfaccia grafica è presentata in Figura 5.6.

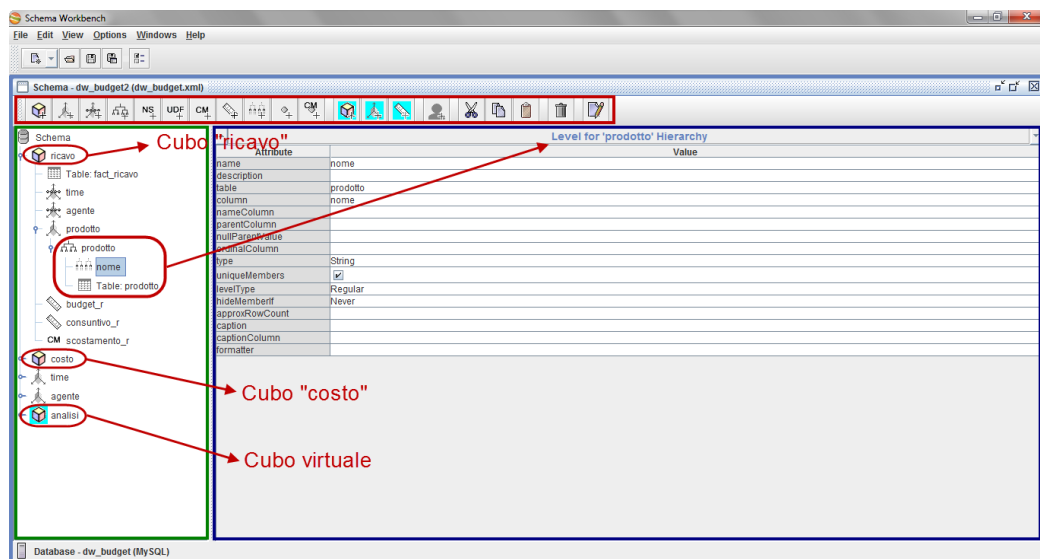


Figura 5.6: Interfaccia dello strumento Schema Workbench

L'area evidenziata in rosso mostra i principali passi da utilizzare nella progettazione. Nell'elenco sono forniti elementi OLAP fondamentali che danno origine alla definizione di uno schema: il cubo, le dimensioni, le misure, le misure calcolate, le gerarchie ecc. Ogni volta che si seleziona ciascuno dei singoli passi, questo va ad integrare la struttura dello schema presentata nell'area evidenziata in verde. La struttura ha una forma ad albero e mostra,

secondo una rappresentazione a cascata, i singoli cubi e gli elementi che la compongono. In figura sono evidenziati i due cubi definiti nel progetto ed il cubo virtuale, con particolare evidenza sul dettaglio del cubo *ricavo*. Per ciascun elemento selezionato e incorporato nella struttura se ne devono definire le caratteristiche principali mostrate nell'area blu (nell'esempio sono riportate le caratteristiche ed i parametri di configurazione per la dimensione *Prodotto*). Tra i vari parametri, si deve inserire il nome della tabella che mantiene le informazioni desiderate (in questo caso la tabella del data warehouse si chiama prodotto come la dimensione stessa), il nome del campo della stessa tabella che si intende recuperare in fase di visualizzazione nelle analisi (sempre in questo caso si va a recuperare il campo nome dalla tabella prodotto), il tipo del campo (se si tratta di elementi di tipo stringa, intero, ecc.), il nome del passo stesso, una breve descrizione sull'utilità e la funzione del passo all'interno dello schema ecc.

Una volta che lo schema è definito e assente da errori, fatti gli opportuni test di controllo, può essere pubblicato all'interno di una macchina server che deposita il file XML corrispondente nel repository (già discusso nel Capitolo precedente) e reso quindi condivisibile per gli utenti.

JPivot

Continuando la rassegna degli strumenti elencati precedentemente, si discute ora l'interfaccia grafica utilizzata dagli utenti per la visualizzazione dei risultati, che prende il nome di *JPivot*. Questo strumento è da considerarsi una libreria di tag JSP personalizzati per interagire con cubi OLAP, grazie al quale si rende possibile l'esecuzione di tipiche operazioni di navigazione degli strumenti OLAP quali: *slice* e *dice*, *roll up* e *drill down*, *pivot*, il tutto in modo semplice e intuitivo, all'insaputa dell'utente finale, che semplicemente utilizza gli strumenti messi a disposizione e compie operazioni di manipolazione del cubo direttamente sulle tabelle visualizzate.

Per la visualizzazione di un cubo di analisi, l'utente deve avviare la procedura *New Analysis View* presente nell'interfaccia grafica della console utente di Pentaho, come mostrato in Figura 5.7. Successivamente il sistema va a recuperare tutti gli schemi depositati nel repository e tramite l'apposito box, presenta la scelta dello schema desiderato. Solo dopo aver definito lo schema

si può scegliere il cubo da visualizzare, che nel caso riportato, è quello inerente ai ricavi aziendali. La Figura 5.7 mostra la sequenza dei passi descritti. A questo punto l'utente può sfruttare le potenzialità dell'interfaccia grafica JPivot, integrata nella user console, nella manipolazione del cubo scelto, mostrata nell'area di colore rosso in Figura 5.8.

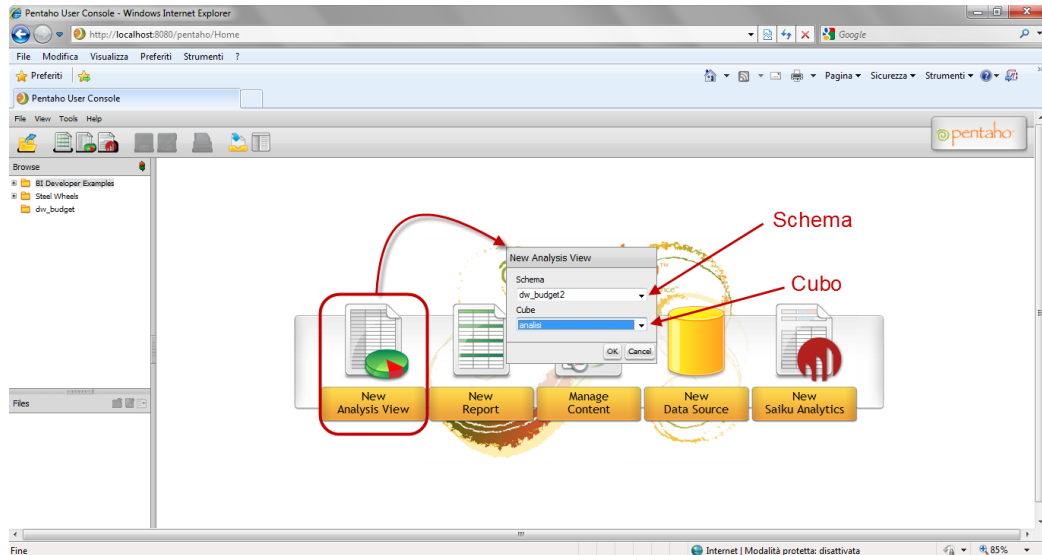


Figura 5.7: Interfaccia Pentaho user console

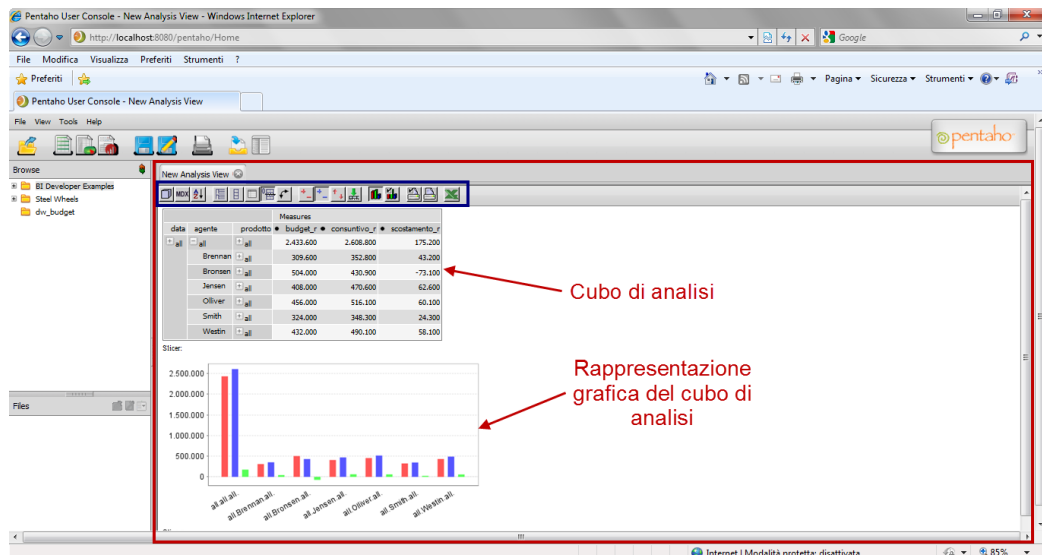


Figura 5.8: Interfaccia di analisi JPivot

JPivot fornisce alcuni strumenti utili per la consultazione dei dati rappre-

sentati nel cubo, riportati all'interno dell'area evidenziata in blu. Tra questi, spicca il *navigatore OLAP* di Figura 5.9.

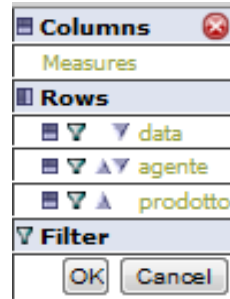


Figura 5.9: Esempio di navigatore OLAP in JPivot

Tramite l'utilizzo di questo strumento l'utente ha piena libertà di scegliere le impostazioni base di visualizzazione del cubo; vale a dire la scelta delle misure che si vuole osservare, la selezione di alcune o tutte le dimensioni di analisi presenti nel cubo, la possibilità di filtrare i risultati visualizzati sulla base di alcuni valori degli attributi dimensionali ecc.

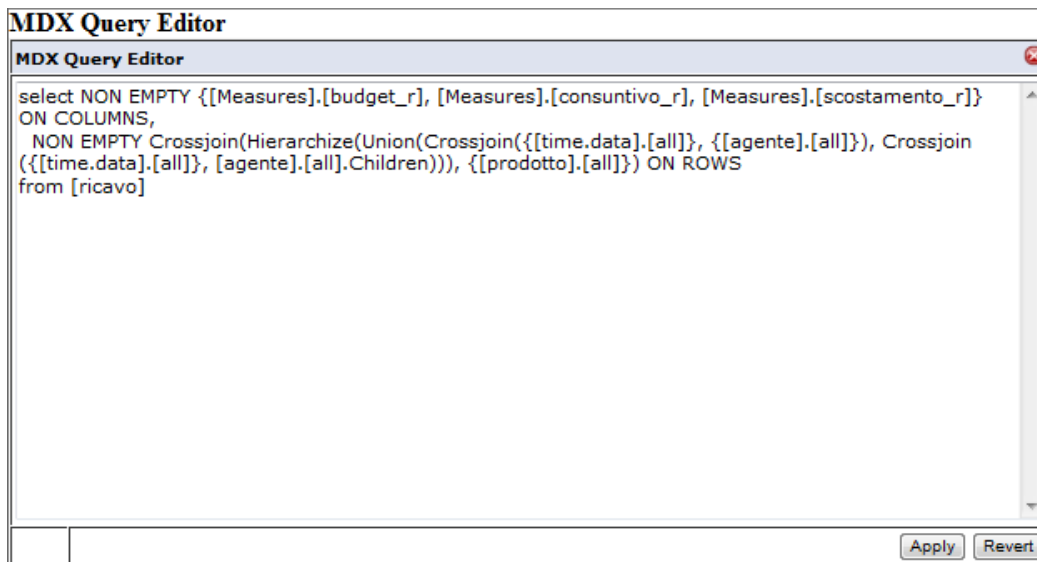


Figura 5.10: Esempio di MDX editor in JPivot

Altri elementi utili sono il *visualizzatore di grafici*, come mostrato nell'esempio di Figura 5.8, che da una rappresentazione grafica dei dati che in quel momento si stanno manipolando nel cubo e l'*MDX editor* che invece resti-

tuisce la corrispondente interrogazione, secondo la sintassi MDX (come mostrato in Figura 5.10), che il sistema esegue sulla base di dati per recuperare le informazioni visualizzate.

Nel complesso, il compito principale dell'interfaccia Jpivot è quello di provvedere a tradurre le richieste pervenute dall'utente in query MDX che poi verranno inoltrate al motore Mondrian ROLAP il quale, sfruttando una connessione JDBC, recupererà i dati sul data warehouse, ritrasmettendoli poi in forma grafica sull'interfaccia stessa.

Saiku

Anche se JPivot continua ad essere lo strumento principale di analisi dei cubi OLAP predefinito nella suite Pentaho C.E., ciò non toglie che tra i vari punti di debolezza si riscontra una scarsa incisività visiva per l'utente che lo utilizza e un certo grado di obsolescenza rispetto ad altri strumenti. Ne consegue che la piattaforma Pentaho permette agli utenti di integrare nuovi strumenti per la business intelligence, tipicamente open source e facilmente reperibili sul web. Uno di questi strumenti è il plug-in *Saiku*.⁴

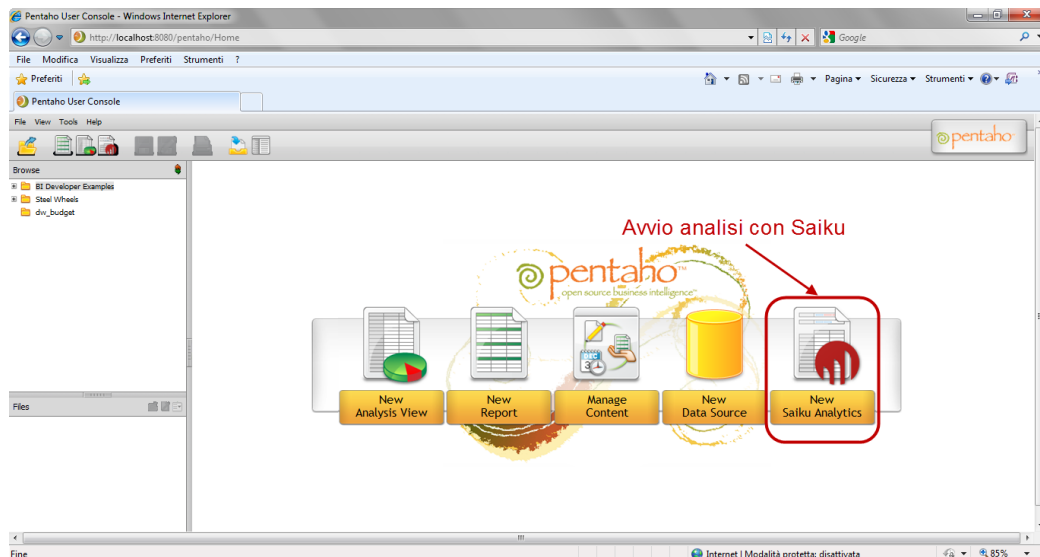


Figura 5.11: Interfaccia Pentaho user console

⁴Per i dettagli sull'installazione e integrazione di Saiku sulla user console di Pentaho C.E. si rimanda il lettore alla lettura della documentazione riportata sul sito web: <http://www.sourceforge.net/projects/saiku/>

Saiku rappresenta una valida alternativa allo strumento JPivot, godendo di un'interfaccia utente più intuitiva e visivamente più chiara e incisiva, oltre alla maggior semplicità di utilizzo nella generazione delle analisi desiderate. La modalità di funzionamento è pressochè identica a quella di JPivot, pertanto anche Saiku raccoglie le informazioni che dovranno essere visualizzate a video recuperando i file XML depositati nel repository definiti con Schema Workbench.

Come per gli altri strumenti presentati precedentemente, si riporta un classico esempio di utilizzo dello strumento Saiku per le analisi sul cubo OLAP dei ricavi aziendali.

In Figura 5.11 è evidenziato il bottone di avvio dello strumento Saiku mantenuto sulla pagina principale della user console di Pentaho. Per iniziare una nuova sessione di analisi di un cubo OLAP, l'utente è tenuto ad avviare la procedura *New Saiku Analytics*, come mostrato in figura.

Una volta che la procedura è stata avviata, Pentaho propone all'utente la nuova interfaccia evidenziata nell'area di colore rosso mostrata in Figura 5.12.

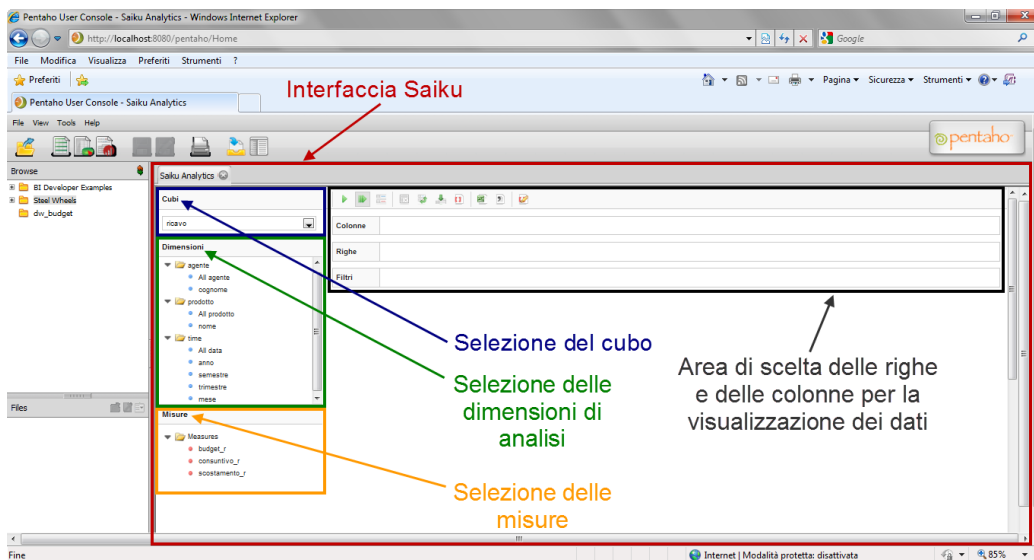


Figura 5.12: Interfaccia di analisi Saiku

Essa si distingue in quattro sotto aree fondamentali:

1. Area *blu*: propone un menù a tendina per la scelta del cubo di interesse per il quale si intende eseguire l'analisi. Rappresenta il primo passo

nell'utilizzo di Saiku, dal momento che i dati che verranno riportati sulle altre aree, sono caricati dinamicamente in funzione dello schema e del relativo cubo scelto. In questo esempio il cubo scelto è quello dei ricavi aziendali.

2. Area *verde*: propone l'insieme delle dimensioni di analisi riportate nel cubo, distinguendo per ciascuna di esse i singoli attributi dimensionali. In questo esempio le dimensioni sono Agente, Prodotto e Data ciascuna con i rispettivi attributi.
3. Area *gialla*: propone le misure di interesse mantenute nella tabella dei fatti, sia le misure calcolate in fase di sviluppo della struttura del cubo. In questo esempio sono riportati i valori dei ricavi a budget ed a consuntivo e il rispettivo scostamento.
4. Area *nera*: propone all'utente la possibilità di costruire la tabella di visualizzazione dei risultati a sua indiscrezione, dando la possibilità di gestire a piacimento le informazioni che dovranno essere mantenute sulle righe e quelle sulle colonne. Inoltre esiste un terzo campo che dà la possibilità di filtrare i risultati mostrati in tabella, sulla base di uno o più valori di attributi di una o più dimensioni di analisi. In questo esempio si riportano sulle colonne le tre misure indicate e sulle righe gli agenti di vendita, eseguendo un filtro sull'attributo anno della dimensione Data.

Nelle Figure 5.13 e 5.14 si propongono i passi per la costruzione di una tabella per l'analisi dei dati del cubo *ricavo*.

Sfruttando la classica tecnica di trascinamento degli oggetti sullo schermo (nota col termine *drag and drop*), l'utente seleziona le misure di interesse riportate nel menù a sinistra dell'interfaccia (area gialla) e le trascina nel menù riportato in alto nell'interfaccia (area nera), rispettivamente sulla sezione marcata come riga o colonna. Nell'esempio le misure *budget_r*, *consuntivo_r* e *scostamento_r* sono riportate sulle colonne della tabella. Analogamente si effettua la stessa operazione sfruttando gli attributi delle dimensioni di analisi riportati anch'essi nel menù a sinistra dell'interfaccia (area verde). Nell'esempio si utilizza l'attributo *cognome* della dimensione Agente riportato sulle righe della tabella. In concomitanza con i passi effettuati dall'utente, il sistema genera dinamicamente la tabella corrispondente mostrandola in

posizione centrale nell'interfaccia, come evidenziato in Figura 5.13.

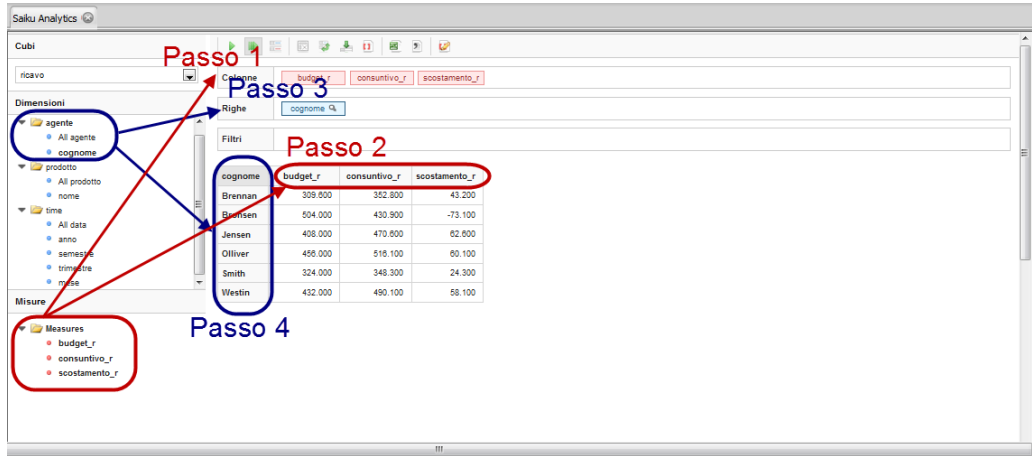


Figura 5.13: Impostazione righe e colonne della tabella

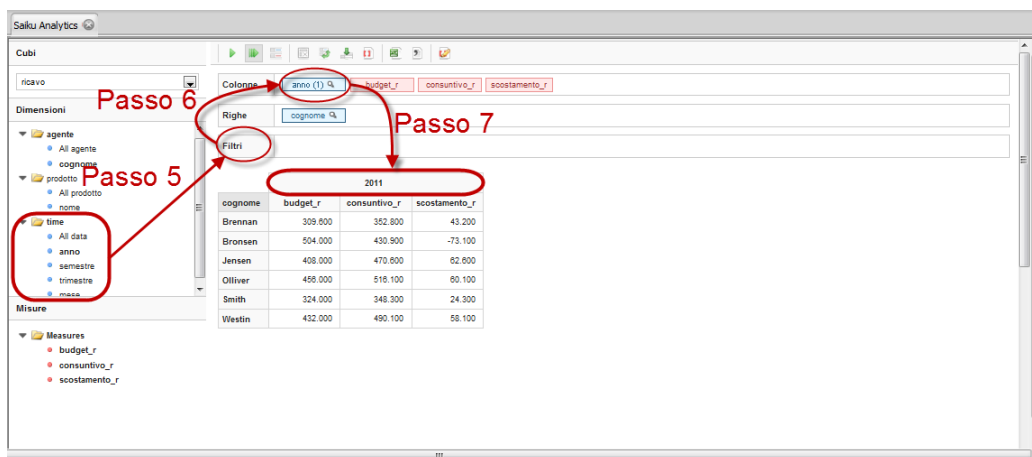


Figura 5.14: Impostazione filtro sulla dimensione di analisi

Supponendo poi di voler aggiungere un filtro su un qualsiasi attributo di analisi, l'utente dovrà, sempre sfruttando la tecnica del drag and drop, trascinare l'attributo nel campo *filtro* riportato nell'area nera. A questo punto lo strumento Saiku, sfruttando una nuova finestra, presenterà la scelta del valore dell'attributo sul quale eseguire il filtro fra tutti i valori disponibili. Eseguite le opportune operazioni di configurazione del filtro, l'attributo dovrà nuovamente essere trascinato sul campo *riga* o *colonna* dell'area evidenziata in nero, qualora si desidera mostrare il valore direttamente sulla tabella.

In Figura 5.14 si mostra la sequenza dei passi utili per l'inserimento di un filtro sul valore 2011 dell'attributo *anno* nella dimensione Data, riportato poi sulle colonne della tabella mostrata al centro dell'interfaccia.

Analogamente a quanto detto sullo strumento JPivot, anche in questo caso Saiku fornisce alcuni strumenti di utilità aggiuntivi. Tra i più importanti si trova l'elemento *Scambia assi* che permette l'inversione reciproca degli elementi ripartiti sulle righe e sulle colonne, il *Drill through current query* che esegue un drill down per tutti i valori degli attributi che fanno parte del cubo, il *Visualizza MDX* che mostra la query nella sintassi MDX che genera i valori riportati nella tabella visualizzata, gli elementi *Esporta XLS* e *Esporta CSV* che creano rispettivamente un nuovo foglio di calcolo Excel o file CSV contenenti i dati riportati sulla tabella di analisi.

Negli esempi di report che verranno presentati nel Capitolo successivo, le tabelle pivot mostrate sono state create utilizzando questo strumento.

5.3 Esempi di analisi dei dati

Nelle Sezioni che seguono si presentano i modelli multidimensionali a cubo definiti per questo progetto (utilizzando l'interfaccia utente JPivot), distinguendo alcuni esempi che mostrano le possibili analisi che un utente può effettuare, sia sui dati di costo che sui dati di ricavo.

5.3.1 Analisi sui dati di costo

In Figura 5.15 è riportato un classico esempio di utilizzo del cubo per la consultazione e il confronto dei valori delle misure utilizzando la gerarchia nella dimensione Data.

Livello massimo di aggregazione

				Measures		
data	agente	sede	voce costo	● budget_c	● consuntivo_c	● scostamento_c
+ all	+ all	+ all	+ all	2.417.080	2.097.970	-319.110

Livello di aggregazione per anno

				Measures		
data	agente	sede	voce costo	● budget_c	● consuntivo_c	● scostamento_c
all	all	all	all	2.417.080	2.097.970	-319.110
2011	all	all	all	2.417.080	2.097.970	-319.110

Livello di aggregazione per semestre

				Measures		
data	agente	sede	voce costo	● budget_c	● consuntivo_c	● scostamento_c
all	all	all	all	2.417.080	2.097.970	-319.110
2011	all	all	all	2.417.080	2.097.970	-319.110
S1	all	all	all	1.223.290	1.187.660	-35.630
S2	all	all	all	1.193.790	910.310	-283.480

Livello di aggregazione per trimestre

				Measures		
data	agente	sede	voce costo	● budget_c	● consuntivo_c	● scostamento_c
all	all	all	all	2.417.080	2.097.970	-319.110
2011	all	all	all	2.417.080	2.097.970	-319.110
S1	all	all	all	1.223.290	1.187.660	-35.630
S2	all	all	all	1.193.790	910.310	-283.480
T3	all	all	all	572.850	529.680	-43.170
T4	all	all	all	620.940	380.630	-240.310

Livello di aggregazione per mese

				Measures		
data	agente	sede	voce costo	● budget_c	● consuntivo_c	● scostamento_c
all	all	all	all	2.417.080	2.097.970	-319.110
2011	all	all	all	2.417.080	2.097.970	-319.110
S1	all	all	all	1.223.290	1.187.660	-35.630
S2	all	all	all	1.193.790	910.310	-283.480
T3	all	all	all	572.850	529.680	-43.170
Luglio	all	all	all	198.510	194.490	-4.020
Agosto	all	all	all	179.980	158.660	-21.320
Settembre	all	all	all	194.360	176.530	-17.830
T4	all	all	all	620.940	380.630	-240.310

Figura 5.15: Esempio di utilizzo del modello multidimensionale

L'interfaccia utente propone inizialmente il massimo livello di aggregazione dei dati, in questo caso sono riportate le somme complessive dei costi azien-

dali sia a budget che a consuntivo. La terza misura, riportante il valore dello scostamento complessivo, è definita come differenza delle due precedenti e calcolata in questa fase di analisi, come discusso nel Capitolo 3. Il valore *all* associato alle dimensioni considerate indica la notazione “per tutti i valori dell’attributo dimensionale considerato”. Pertanto nell’esempio riportato è mostrato l’ammontare complessivo dei costi a: budget, consuntivo e relativo scostamento per l’anno in corso 2011 per tutte: le sedi, le voci di costo e gli agenti. Ipotizzando di voler scendere ulteriormente nel dettaglio, in figura è riportata la successiva distinzione dell’anno 2011 nel primo semestre (S1) e nel secondo (S2). A questo punto l’ammontare complessivo dei valori delle misure verrà automaticamente partizionato tra i due nuovi attributi dimensionali della gerarchia. Passando poi attraverso la distinzione in trimestri dell’anno corrente (nell’esempio è mostrato il terzo trimestre T3) si può raggiungere il livello di aggregazione minimo rappresentato dai singoli mesi che determinano il livello di granularità del fatto *costo a budget e consuntivo* specificato in fase di progettazione del data mart (Figura 3.5).

Nel caso di analisi più complesse e dettagliate è possibile, tramite opportune manipolazioni del cubo, eseguire interrogazioni che sfruttano alcuni operatori base, quali *roll-up* e *drill-down*, rispettivamente diminuendo o aumentando il numero delle dimensioni di analisi. In parte questi operatori sono già stati osservati nell’esempio di Figura 5.15 nel momento in cui il livello di dettaglio dell’analisi aumenta aggregando di volta in volta per attributi dimensionali più specifici (ad esempio quando si passa dall’aggregazione per trimestre a quella più specifica per mese). In generale gli operatori menzionati si possono definire come segue [Albano 09a]:

- *Drill-down*: è l’operatore che permette di aumentare il livello di dettaglio dell’analisi considerando più dimensioni o aggregando per attributi dimensionali più specifici (ad esempio quando si passa dall’aggregazione sulla misura importo a budget raggruppata per trimestre all’aggregazione sulla misura importo a budget raggruppata per mese). Lo si può considerare come l’operatore inverso al *roll-up*. Nella rappresentazione relazionale dei dati questa operazione si esprime con il GROUP BY, più precisamente aggiungendo un attributo nella SELECT e nella GROUP BY (ad esempio aggiungendo il mese).

agente	data	sede	voce costo	Measures		
				● budget_c	● consuntivo_c	● scostamento_c
all	all	all	all	2.417.080	2.097.970	-319.110
Brennan	all	all	all	329.600	271.780	-57.820
	2011	all	all	329.600	271.780	-57.820
	S1	all	all	166.400	150.925	-15.475
	T1	all	all	82.100	74.635	-7.465
	T2	all	all	84.300	76.290	-8.010
	S2	all	all	163.200	120.855	-42.345
	T3	all	all	78.100	70.020	-8.080
	T4	all	all	85.100	50.835	-34.265
Bronsen	all	all	all	457.690	413.540	-44.150
Fittizio	all	all	all	136.000	100.900	-35.100
Jensen	all	all	all	365.840	335.060	-30.780
Olliver	all	all	all	419.000	371.770	-47.230
Smith	all	all	all	322.310	265.675	-56.635
Westin	all	all	all	386.640	339.245	-47.395

Figura 5.16: Esempio di raggruppamento su due dimensioni

Un possibile esempio di utilizzo dell'operatore drill-down è riportato in Figura 5.16. In questo caso l'interrogazione effettuata calcola il valore della funzione di aggregazione somma applicata per le misure budget e consuntivo (per l'agente Brennan), raggruppando i dati sia sulla dimensione Agente che sulla dimensione Data. Il risultato così ottenuto evidenzia l'ammontare delle due misure di un certo agente per un certo orizzonte temporale, favorendo ad esempio il confronto dei risultati ottenuti da quest'ultimo nei quattro trimestri dell'anno.

- *Roll-up*: è l'operatore che permette di raggruppare i dati su alcune dimensioni e calcola il valore di una funzione di aggregazione applicata ad una misura. Questo operatore diminuisce il livello di dettaglio dell'analisi riducendo il numero delle dimensioni. Se sugli attributi delle dimensioni sono definite delle gerarchie, il roll-up è utilizzato per passare da un'aggregazione su una misura con un certo livello di raggruppamento su una dimensione, ad una nuova aggregazione sulla medesima misura contenente un raggruppamento con un livello di dettaglio minore (ad esempio quello che accade quando si passa dall'aggregazione sulla misura importo a budget raggruppata per mese all'aggregazione sulla misura importo a budget raggruppata per trimestre). Nella rappresentazione

relazionale dei dati questa operazione si esprime con il GROUP BY, più precisamente togliendo un attributo nella SELECT e nella GROUP BY (ad esempio togliendo il mese ma lasciando il trimestre).

5.3.2 Analisi sui dati di ricavo

Analogamente a quanto detto in precedenza, gli stessi risultati sono stati ottenuti per i dati che riguardano i ricavi aziendali. In questo caso il cubo OLAP è realizzato da una rappresentazione multidimensionale delle informazioni contenute nel data mart *ricavo a budget e consuntivo* (Figura 3.6).

Nell'esempio di Figura 5.17 il valore della funzione di aggregazione somma applicata alle misure importo a budget e importo a consuntivo viene riportata sfruttando un raggruppamento sui dati della dimensione Agente.

			Measures		
data	agente	prodotto	● budget_r	● consuntivo_r	● scostamento_r
+ all	- all	+ all	2.433.600	2.608.800	175.200
	Brennan	+ all	309.600	352.800	43.200
	Bronsen	+ all	504.000	430.900	-73.100
	Jensen	+ all	408.000	470.600	62.600
	Olliver	+ all	456.000	516.100	60.100
	Smith	+ all	324.000	348.300	24.300
	Westin	+ all	432.000	490.100	58.100

Figura 5.17: Esempio di analisi dei dati per agente

5.3.3 Analisi complessiva dei dati

Il sistema di analisi utilizzato permette agli sviluppatori di poter definire delle viste su modelli multidimensionali a cubo virtuali (chiamati *virtual cube*). La possibilità della loro progettazione deriva dalla necessità di dover talvolta offrire delle analisi su dati mantenuti su data mart differenti.⁵

Più precisamente, il frequente uso dei cubi virtuali nelle applicazioni reali, avviene per risolvere almeno una delle due seguenti problematiche:

⁵Maggiori dettagli sull'utilizzo dei cubi virtuali e sulla loro implementazione si possono ricercare sulla pagina web <http://mondrian.pentaho.com/>

1. Si vogliono eseguire analisi su dati appartenenti a distinte tabelle dei fatti caratterizzate da granularità differenti (ad esempio quando la granularità di un fatto è il giorno mentre la granularità dell'altro è il mese).
2. Si vogliono eseguire analisi su dati appartenenti a distinte tabelle dei fatti caratterizzate da diverse dimensioni associate (ad esempio quando le dimensioni di un fatto sono: Prodotto, Data e Cliente, mentre le dimensioni dell'altro sono: Prodotto, Data e Magazzino).

Il cubo virtuale offre una possibile soluzione a questi due problemi presentando i risultati ottenuti ad un generico utente che principalmente non si occupa o non si cura di come effettivamente i dati siano strutturati. Nel caso di studio affrontato in questa tesi, il cubo virtuale ha permesso la soluzione del problema presentato al punto 2.

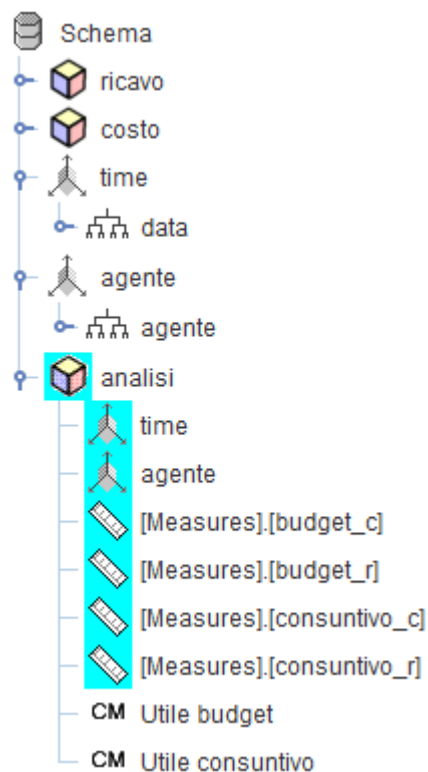


Figura 5.18: Schema di struttura del cubo virtuale

In Figura 5.18 è mostrato lo schema di composizione del cubo virtuale denominato Analisi. Per poter usufruire di questo strumento, occorre in primo

luogo mettere a disposizione i cubi progettati per le analisi dei dati appartenenti ai fatti interessati sui quali si vuole definire la vista virtuale, che nell'esempio prendono il nome di Costo e Ricavo. Solo successivamente si vanno a definire gli elementi dimensioni (*virtual cube dimension*) e le misure (*virtual cube measure*) per la vista virtuale. Tipicamente l'elemento dimensione virtuale importa una o più dimensioni da uno dei cubi sui quali si genera la vista e in modo analogo si importano le misure sfruttando l'elemento misura virtuale. Nel caso riportato, le dimensioni Data e Agente costituiscono gli unici due elementi di raggruppamento dei valori delle misure in quanto sono le uniche dimensioni comuni ai due fatti considerati.

Per quanto riguarda le misure, oltre ad importare quelle appartenenti ai cubi considerati, è possibile definire nuove misure calcolate sfruttando l'elemento *calculate member*. In questo caso si fa riferimento all'utile a budget e all'utile a consuntivo, calcolati come differenza delle somme sugli importi di costo e ricavo sia a budget che a consuntivo.

Il cubo virtuale, la cui struttura è riportata in Figura 5.18, permette quindi di poter eseguire analisi dei dati risiedenti sui due data mart progettati, favorendo una visione di insieme delle misure riportate. Un esempio di analisi è riportato in Figura 5.19.

		Measures					
data	agente	● budget_c	● budget_r	● consuntivo_c	● consuntivo_r	● Utile budget	● Utile consuntivo
all	all	2.417.080	2.433.600	2.097.970	2.608.800	16.520	510.830
2011	all	2.417.080	2.433.600	2.097.970	2.608.800	16.520	510.830
S1	all	1.223.290	1.216.800	1.187.660	1.500.800	-6.490	313.140
S2	all	1.193.790	1.216.800	910.310	1.108.000	23.010	197.690
T3	all	572.850	608.400	529.680	610.100	35.550	80.420
Luglio	all	198.510	202.800	194.490	198.100	4.290	3.610
Agosto	all	179.980	202.800	158.660	134.300	22.820	-24.360
Settembre	all	194.360	202.800	176.530	277.700	8.440	101.170
	Brennan	26.300	25.800	25.330	44.000	-500	18.670
	Bronsen	36.350	42.000	34.930	51.000	5.650	16.070
	Fittizio	10.000		1.060		-10.000	-1.060
	Jensen	30.100	34.000	29.440	46.000	3.900	16.560
	Olliver	34.500	38.000	30.140	45.600	3.500	15.460
	Smith	25.820	27.000	25.090	43.000	1.180	17.910
Westin	31.290	36.000	30.540	48.100	4.710	17.560	
T4	all	620.940	608.400	380.630	497.900	-12.540	117.270

Figura 5.19: Esempio di analisi dei dati sul cubo virtuale

Si nota come sia possibile eseguire un'analisi comparativa sull'ammontare

complessivo degli importi di costi e ricavi sia a budget che a consuntivo, per agente e per data. Nel caso riportato l'analisi è eseguita per tutti gli agenti nell'ultimo mese del terzo trimestre. Purtroppo questa tecnica consente analisi basate principalmente sulle dimensioni comuni ad entrambi i fatti considerati, motivo per cui il cubo virtuale riporta solo la dimensione Agente e Data, perdendo quindi parte dei punti di vista osservati nei casi precedenti (nello specifico si perde il punto di vista per Sede, per Voce di costo e per Prodotto).

Vi è piena libertà, da parte degli sviluppatori del sistema di analisi, di lasciare al committente la manipolazione dei modelli a cubo, una volta che questi sono stati progettati, tramite opportune interfacce grafiche. In alternativa o in aggiunta a questa scelta è possibile creare e mettere a disposizione degli utenti incaricati opportuni documenti di reportistica basati sul recupero delle informazioni generate in questa fase di progettazione. Nel Capitolo successivo si osservano alcuni esempi dei report definiti.

5.4 Osservazioni conclusive

Complessivamente, gli strumenti utilizzati per la progettazione dei modelli multidimensionali a cubo, sono: il motore OLAP di Pentaho (Mondrian ROLAP), l'interfaccia grafica di costruzione dei cubi OLAP (Schema Workbench) e l'interfaccia utente di visualizzazione e manipolazione dei cubi JPivot o alternativamente lo strumento Saiku. Brevemente si riassumono i passaggi che determinano le relazioni tra i tre strumenti:

1. L'utente progettista costruisce la struttura del o dei cubi sfruttando l'interfaccia grafica Schema Workbench. Sarà compito di quest'ultima tradurre i passaggi in un file XML che determina le caratteristiche dello schema principale e dei cubi contenuti. Testata la validità e la correttezza dei risultati, il file viene depositato nella directory repository rendendolo così condivisibile per gli utenti che utilizzano la Pentaho user console.
2. L'utente utilizzatore effettua una classica connessione HTTP da terminale allo strumento JPivot, integrato nella console utente di Pentaho.

Il cubo scelto dall'utente ora può essere visualizzato, sfogliato e manipolato con operazioni di drill down direttamente sulla tabella pivot mostrata nell'interfaccia.

3. JPivot raccoglie ogni singolo passaggio di manipolazione del cubo che l'utente effettua a livello grafico e lo traduce in una query MDX che verrà inviata al motore Mondrian ROLAP.
4. Mondrian ROLAP, ricevuta la query MDX, provvede alla traduzione della stessa in una o più query SQL, inoltrandola poi alla base di dati interessata per mezzo di connessioni JDBC.
5. A sua volta l'RDBMS esegue la query ricevuta, estrae i risultati dalle tabelle interessate e restituisce le informazioni al motore Mondrian ROLAP.
6. Mondrian ROLAP processa i risultati ricevuti traducendoli in *result-set* multidimensionali.
7. Questo risultato multidimensionale può ora essere inoltrato allo strumento JPivot che produce una pagina web HTML per la visualizzazione dei dati. La pagina HTML, inviata al browser dell'utente, ne permette la visualizzazione finale.

Una tematica interessante, che merita un'attenzione particolare, riguarda il connubio tra la progettazione dei data mart secondo una struttura logica a stella e il motore Mondrian. Questi due aspetti sono fortemente legati fra loro da relazioni e vincoli progettuali.

In primo luogo si ricorda che Mondrian è un tipico motore ROLAP, questo implica che i dati risiedono tutti su un database relazionale su disco, pertanto ad ogni richiesta di informazione, lo strumento avvia una procedura di accesso al database fisico con conseguenti decrementi prestazionali rispetto a strumenti MOLAP che invece mantengono i dati su appositi spazi di memoria dedicati. D'altro canto occorre puntualizzare che Mondrian

sfrutta anche un meccanismo di caching interno che di fatto lo rende un motore ibrido HOLAP. Così definito, lo strumento non necessita la creazione di ulteriori sovrastrutture per il mantenimento dei dati e quindi le informazioni risiedenti nel data warehouse risultano sufficienti per la creazione di cubi OLAP senza l'utilizzo di particolari viste, evitando eventuali ridondanze dell'informazione. Alternativamente, qualora si rende indispensabile l'utilizzo di informazioni precalcolate per il miglioramento delle prestazioni, Mondrian consente la definizione di tabelle aggregate tramite lo strumento *Aggregation Designer* (tipicamente utile quando si conosce anticipatamente la natura delle analisi che verranno effettuate dall'utente, ad esempio quando è verificato che il 90% di queste si basano sull'attributo *anno* della dimensione Data è possibile definire delle tabelle preaggregate per quell'attributo).

Da quanto detto, unito alla particolarità che Pentaho C.E., in qualità di strumento di BI open source, non è vincolato a particolari tipologie di DB e architetture, ne deriva che lo strumento Mondrian ROLAP può lavorare indifferentemente con schemi a stella o a fiocco di neve di un data warehouse. La necessità di dover strutturare il data warehouse secondo le caratteristiche di un determinato schema (schemi a stella nel caso del progetto presentato in questa tesi) deriva soprattutto dal forte legame che la struttura ha con le esigenze di analisi sollevate dall'utilizzatore del sistema (l'azienda cliente in questo specifico caso). La questione si sposta quindi non tanto sul fatto di dover progettare uno o più data mart secondo le regole che regolamentano un certo schema, quanto sul come quel determinato schema dovrà essere strutturato affinché sia utilizzabile al meglio dallo strumento Mondrian ROLAP. Nella Sezione 5.2.1, discutendo sull'utilizzo di Schema Workbench e delle relazioni che questo strumento ha con il motore OLAP, si osserva nel dettaglio il file XML automaticamente prodotto e come su questo si faccia frequente uso di tag che etichettano i dati come un fatto, una misura, una dimensione, una gerarchia ecc. Implicitamente, quindi, ne deriva che alcuni strumenti della suite Pentaho C.E. richiedono l'utilizzo di data mart progettati secondo lo schema a stella (o alternativamente a fiocco di neve).⁶

⁶Su versioni successive della suite Pentaho C.E. rispetto a quella usata per questo progetto (versione 3.7) sono integrati nella user console alcuni strumenti che consentono la definizione di cubi OLAP direttamente all'utente finale. In questi strumenti si evidenzia

Un maggior approfondimento merita anche il tema relativo alla necessità di sviluppare delle strutture a cubo per le analisi OLAP delle informazioni. Per ciascun esempio di analisi mostrato in precedenza, sia esso definito con l'interfaccia JPivot o Saiku, i risultati riportati sul terminale sono sempre frutto dell'esecuzione di una determinata query SQL sulla base di dati. Detto ciò, ci si potrebbe domandare perchè non si possa lasciare all'utente finale la scrittura diretta di una interrogazione SQL, anzichè spingere sull'utilizzo di strumenti dedicati che talvolta possono risultare obsoleti o di difficile usabilità. Le motivazioni sono molteplici ed in primo luogo risalta il fatto che il motore OLAP della suite Pentaho C.E. riceve interrogazioni nella sintassi MDX, tipicamente più complessa e meno conosciuta rispetto alla tradizionale sintassi dell'SQL. Supponendo però che l'utente utilizzatore del sistema abbia le conoscenze adeguate per scrivere correttamente interrogazioni SQL o MDX, si presume che egli, ogni qual volta abbia bisogno di cambiare le esigenze di analisi dei dati, sia costretto a definire una nuova query.

Per fare un esempio, si supponga di voler visualizzare i ricavi totali per un certo prodotto nell'anno 2011. Questo risultato viene fornito dall'esecuzione di una determinata interrogazione SQL sulla base di dati. Se in periodi successivi, muta l'esigenza di analisi, spostando ad esempio il dettaglio dall'anno 2011 ad un certo mese dell'anno, l'interrogazione precedente non è più valida e pertanto è necessaria la riscrittura di una nuova query diversa dalla precedente.

La necessità di usufruire di strutture come i cubi OLAP, create dagli utenti sviluppatori per l'uso da parte degli utenti finali, evita per quest'ultimi di dover modificare la struttura del sistema in relazione ai cambiamenti delle esigenze di analisi. In altri termini si astrae l'utente finale dal fatto di dover necessariamente comprendere come deve essere definita l'interrogazione utile per una determinata analisi (è lo strumento utilizzato che si occupa di costruire l'interrogazione adatta e non l'utente); si astrae l'utente dal fatto di doversi interessare della locazione fisica dei dati e su come è fatta la base di dati che li contiene.

maggiormente la necessità di dover usufruire di data mart definiti secondo schemi a stella perchè i concetti di dimensione, fatto, misura e gerarchia sono espressamente richiesti al momento della progettazione.

Tutti questi elementi concorrono poi verso la proprietà di riusabilità nel tempo di ciò che viene mantenuto nell'intero sistema di analisi. Questa è anche una delle caratteristiche principali che si ritrovano nei molteplici strumenti per la business intelligence, ovvero la capacità di astrazione dell'utente da tutti i meccanismi e le metodologie utili a fornire un determinato risultato, la capacità di centralizzare l'informazione e renderla di facile accesso all'utenza ecc.

Un altro tema di interesse è la questione che riguarda la visualizzazione delle interrogazioni sulla base di dati. Negli esempi in cui si mostrano le modalità di utilizzo degli strumenti, si evidenzia il fatto che questi riportano sull'interfaccia utente un tool specifico per la visualizzazione dell'interrogazione MDX. Per la visualizzazione della stessa query in SQL invece non è previsto alcun elemento in grado di fornirla a livello di interfaccia grafica. Ciò non toglie che anche la query SQL possa essere recuperata, ma allo scopo occorre prima modificare opportuni comandi contenuti in determinati *file di properties* dello strumento Mondrian ROLAP e solo successivamente rintracciare la query all'interno di un particolare *file di log* contenuto in una sotto directory di sistema di Pentaho C.E. Nello specifico il file di log indicato, nominato *mondrian_sql*, è rintracciabile tramite il path `.../pentaho/biserver-ce-3.7.0-stable/biserver-ce/tomcat/bin`.

5.5 Riassunto

- Si tratta l'argomento di analisi dei dati a partire dalle differenze tra sistemi OLTP e OLAP. In seguito si pone l'attenzione sui sistemi OLAP e si riportano alcune regole e definizioni che approfondiscono il tema.
- Si discute il modello multidimensionale a cubo utilizzato nella rappresentazione delle informazioni, entrando nello specifico della progettazione dei cubi multidimensionali attraverso il relativo strumento di BI presente nella Suite Pentaho C.E.
- Per ciascuna tabella dei fatti mostrata nella fase di progettazione del data warehouse si riporta il modello multidimensionale a cubo relativo ed alcuni esempi di analisi dei dati.

-
- Vengono progettati due cubi, il primo che definisce una rappresentazione multidimensionale dei dati di costo (sia a budget che a consuntivo), il secondo che definisce una rappresentazione multidimensionale dei dati di ricavo.
 - Sfruttando le caratteristiche dello strumento si mostra un terzo cubo, il cubo virtuale, rappresentante una nuova vista che offre la possibilità di eseguire analisi su dati mantenuti in distinti data mart. Questo permette di rappresentare su un unico cubo i dati di costo e di ricavo.

Capitolo 6

Il sistema di reporting

La fase conclusiva del progetto riguarda la generazione della documentazione necessaria ai fini informativi che scaturisce dalle analisi definite nel passo precedente. Si osserva nelle Sezioni seguenti l'insieme dei rapporti di sintesi creati accompagnati dalle corrispondenti query SQL eseguite sul data warehouse per l'estrazione dei dati, utilizzando la distinzione in rapporti relativamente semplici, rapporti relativamente complessi e rapporti complessi. Inoltre si dedica una Sezione che mostra la sequenza dei passi utili per la generazione di un rapporto tramite gli strumenti forniti in dotazione nella suite Pentaho C.E. e eventuali tecniche alternative di progettazione. Per concludere si discute sulle potenzialità degli strumenti utilizzati in relazione alle query sviluppate, agli operatori utilizzabili e alle funzioni analitiche necessarie per l'ottenimento dei risultati voluti.

6.1 Gli strumenti della suite Pentaho C.E.

Esistono diverse tipologie di rappresentazione dei report e tra queste, come descritto su [Albano 09a], le più utilizzate sono:

- I *rapporti tradizionali* che organizzano il risultato con colonne di dati, intestazioni e uno o più livelli di dati di riepilogo parziali.
- Le *tabelle a doppia entrata* che mostrano le aggregazioni delle misure rispetto ai valori delle dimensioni lungo gli assi cartesiani. Aggiungendo e togliendo dimensioni si ottengono gli effetti delle operazioni di *roll up* e *drill down*.

- I *grafici* di diversa natura, come ad esempio gli istogrammi, i diagrammi lineari, i diagrammi a torta ecc.

In questa Sezione si mostra l'utilizzo degli strumenti forniti in Pentaho C.E. che permettono la definizione dei rapporti descritti, dedicando maggiore dettaglio alla realizzazione dei grafici. Gli strumenti in oggetto sono:

1. *Report Designer* (strumento messo a disposizione nella suite per la creazione e parametrizzazione di un report con l'impiego di query SQL).
2. *File Xaction* (metodologia alternativa per la creazione dei report).

Report Designer

L'interfaccia grafica messa a disposizione per gli utenti che intendono sviluppare un nuovo rapporto, mostrata in Figura 6.1, permette la creazione, parametrizzazione e pubblicazione dei report.

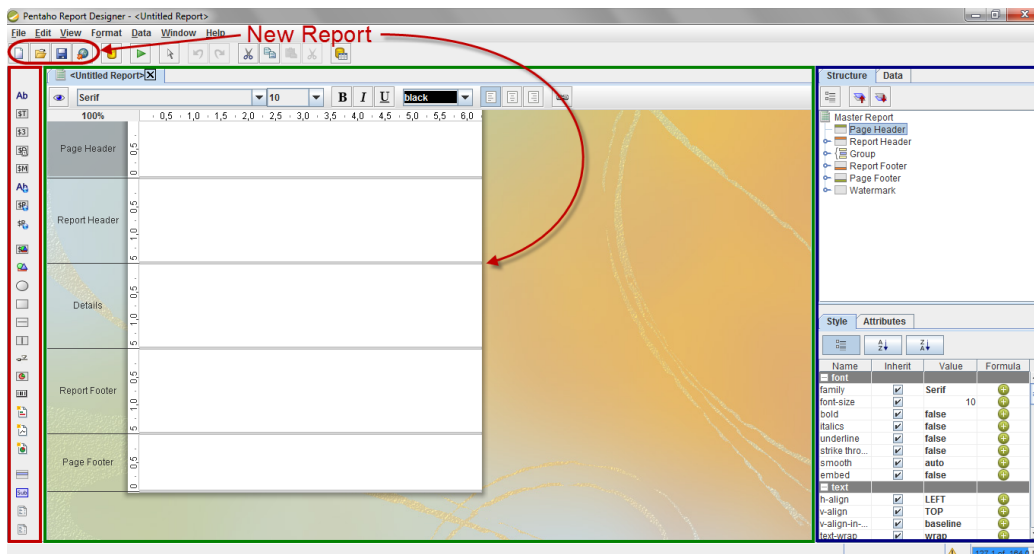


Figura 6.1: Interfaccia dello strumento Report Designer

L'area centrale, evidenziata nel riquadro verde, mostra la struttura fisica del rapporto distinta su cinque fasce principali. La sezione *Page Header* è riservata tipicamente per i contenuti visualizzati in testa ad ogni pagina costituente il report, ad esempio il logo aziendale. Nella sezione sottostante, il *Report Header*, i contenuti inseriti verranno invece visualizzati solamente

nella prima pagina del rapporto e tipicamente si riserva questa area per il titolo del report, ad esempio “rapporto annuale delle vendite per agente”, oltre ad eventuali sintesi dei contenuti che seguono. La sezione *Details* costituisce il corpo centrale del report ed è quella adibita alla visualizzazione dei dati richiamati da una interrogazione sul database, ad esempio i vari importi totali delle vendite riscontrati per ciascun agente. Per concludere, le due sezioni rimanenti, la *Report Footer* e la *Page Footer* sono speculari alle rispettive sezioni mostrate in testa al documento. In particolare, nella prima si inseriscono i contenuti che verranno visualizzati una sola volta a termine del rapporto, ad esempio i totali assoluti riferiti ai dati sopra riportati, mentre nella seconda si inseriscono i contenuti che invece verranno visualizzati a termine di ciascuna pagina che costituisce il report, ad esempio i numeri di pagina ed altre informazioni specifiche dell’azienda.

Il menù di oggetti evidenziato nell’area rossa di Figura 6.1 mantiene l’elenco degli elementi che si vanno a collocare nelle varie sezioni che costituiscono il documento tramite la tecnica del drag and drop già utilizzata per altri strumenti di progettazione. Tra i principali si ritrovano elementi per la visualizzazione di testo statico (ad esempio l’oggetto *label*), elementi per la visualizzazione dei contenuti estratti da una sorgente dati (ad esempio l’oggetto *text-field* e *number-field*), elementi per la visualizzazione di immagini e grafici (ad esempio gli oggetti *image* e *chart*), elementi per l’inserimento di sottoreport (ad esempio l’oggetto *sub-report*) ecc.

Per ciascuno dei molteplici oggetti, una volta inseriti all’interno di una sezione del documento, si possono definire le proprietà che lo caratterizzano e gli stili di formattazione che si desiderano, riportati all’interno dell’area *Style* e *Attributes* mostrata nel riquadro blu in figura. Nello specifico, l’area *Style* mantiene le impostazioni utili alla gestione dell’aspetto di un elemento, l’area *Attributes* invece raccoglie le impostazioni utili alla determinazione del contenuto e del comportamento dell’elemento. A seconda dell’elemento selezionato, i contenuti di impostazione possibili mantenuti sulle due aree possono subire delle variazioni (ad esempio si ha diversità tra un elemento *label* e un *chart*).

Sempre nel riquadro blu, nella parte in alto dell’area denominata *Structure*, viene rappresentato automaticamente la struttura ad albero del report

(la cui radice prende il nome di *Master Report*), distinta per sezioni e su ciascuna di esse sono riportati i vari elementi che ne fanno parte.

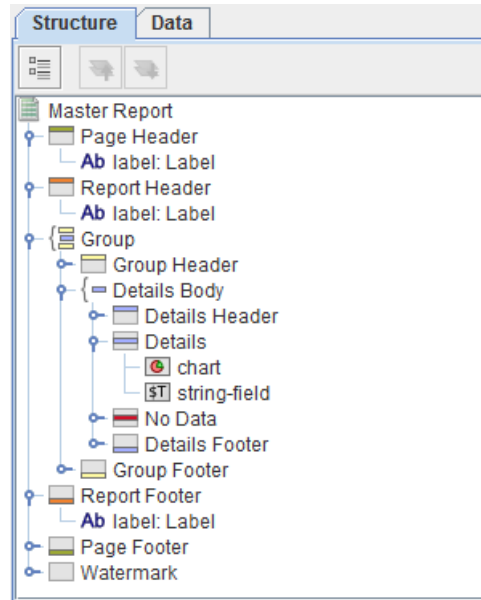


Figura 6.2: Esempio di struttura ad albero del Master Report

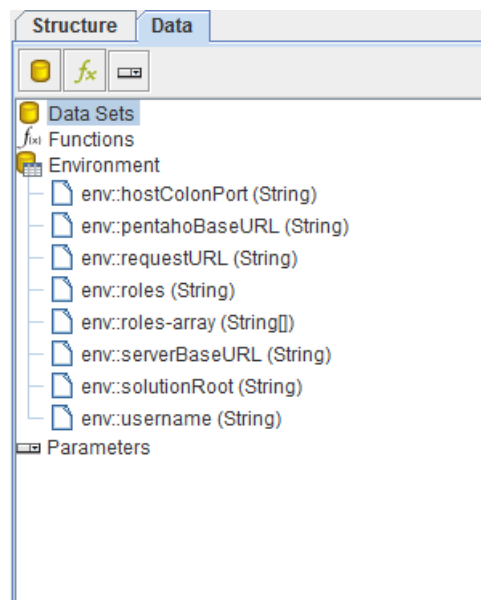


Figura 6.3: Sezione dedicata al recupero dei dati

Ad esempio, in Figura 6.2 si nota come nella sezione Details sono stati inseriti un elemento *chart* e un elemento *string-field*. Questa rappresentazione

fornisce in pratica la struttura gerarchica degli elementi inseriti nel report che si sta progettando.

Oltre alla Structure, sempre nello stesso riquadro, si trova l'area *Data*, mostrata in Figura 6.3 che invece è dedicata alla gestione di tutti i dati inseriti nel report e richiamati da una sorgente dati. Pertanto si vanno a gestire l'interrogazione che restituisce i risultati, eventuali funzioni per ulteriori manipolazioni delle informazioni che si vogliono rappresentare, i parametri ecc. Fatte queste premesse, si definisce ora la sequenza di passi utili per la progettazione di un nuovo rapporto. Inizialmente occorre selezionare la funzione *New Report*, mostrata in Figura 6.1, che propone all'utente la struttura vuota del rapporto. Il secondo passo è dedicato al recupero delle informazioni da una sorgente dati.

In primo luogo occorre specificare la sorgente dati da cui provengono le informazioni che verranno mostrate nel rapporto. Tramite la funzione *Add Data Source*, mostrata in Figura 6.4, si possono scegliere tra varie possibilità: una connessione JDBC, dati provenienti dallo strumento Pentaho Data Integration (discusso nel Capitolo 4), dati provenienti dallo strumento OLAP *Pentaho Analysis*, dati mantenuti su un file XML ecc. In questo esempio si mostra il più classico dei casi, ovvero i dati recuperati tramite una connessione JDBC su una base di dati.

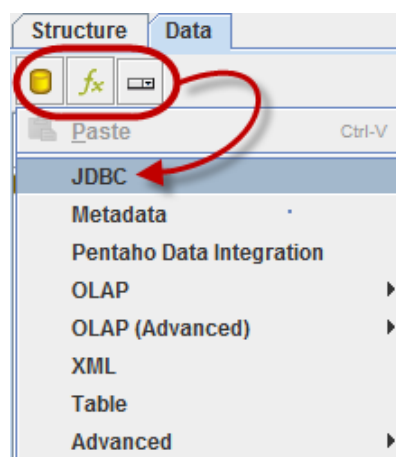


Figura 6.4: Impostazione della sorgente dati

Successivamente, come mostrato in Figura 6.5, si stabiliscono i parametri di connessione alla sorgente dati vera e propria. Se la connessione è già presen-

te nell'elenco *Connections*, mostrato nella prima finestra della figura, allora occorrerà solamente selezionarla e avviare la procedura.

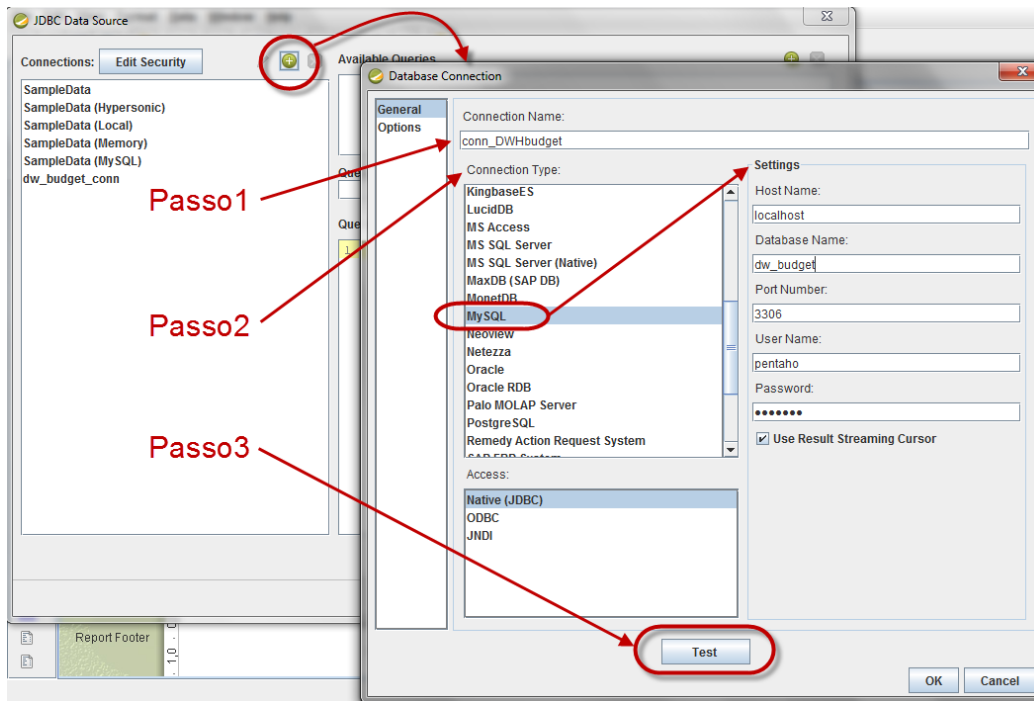


Figura 6.5: Creazione di una connessione al data source

In caso contrario, si deve stabilire una nuova connessione. Per fare ciò si utilizza il bottone *Add a new connection*, evidenziato in rosso in figura. Questa azione avvia la procedura *Database Connection*, con la quale si vanno ad inserire i parametri necessari. I passi utili sono mostrati sequenzialmente in Figura 6.5. Si dichiara un nome alla connessione, nell'esempio è *conn.DWH-budget* (passo 1), si seleziona il tipo di database di riferimento e si inseriscono i parametri utili (il nome dell'*host*, il nome della base di dati, il numero di porta e i parametri di autenticazione per l'accesso), nell'esempio è MySQL (passo 2) ed infine si esegue il test della connessione (passo 3). Se il test ha dato esito positivo, la connessione al database è stata stabilita correttamente e la procedura può terminare. A questo punto Report Designer è connesso alla sorgente dati.

Il passo successivo riguarda la creazione dell'interrogazione che si desidera effettuare sulla sorgente dati di cui ne abbiamo appena stabilito una connessione. A tale scopo Report Designer utilizza lo strumento *SQL Query*

Designer mostrato in Figura 6.6.

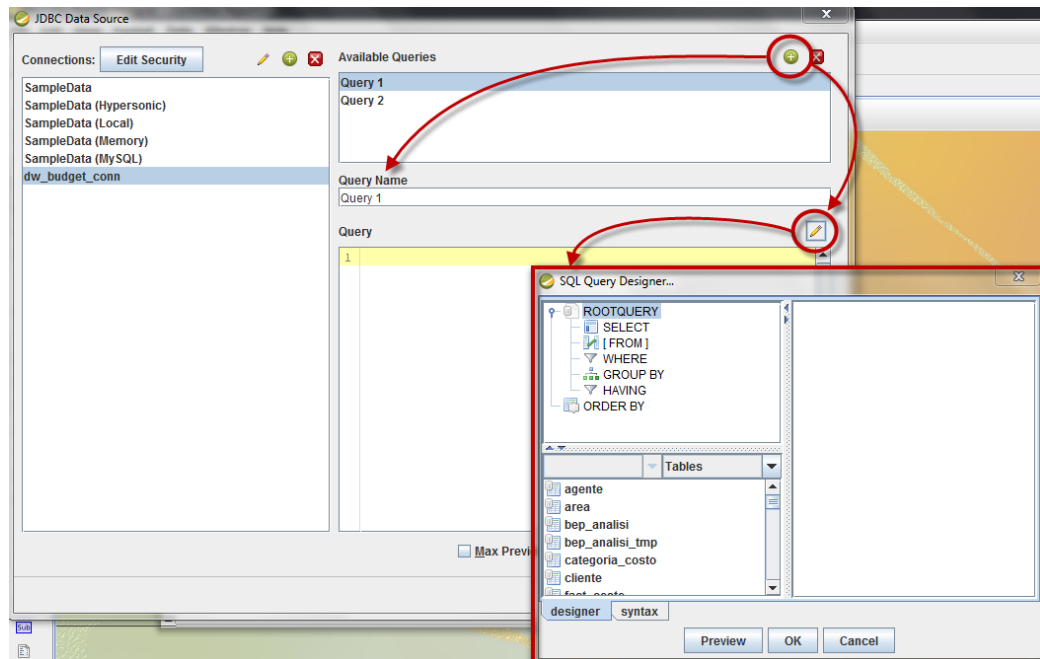


Figura 6.6: Interfaccia di sviluppo SQL Query Designer

Vi si accede mediante l'interfaccia osservata in precedenza, dedicata alla parametrizzazione della sorgente di dati, salvo poi orientarsi verso l'elenco *Available Queries* anzichè nell'elenco *Connections*. Avviata la procedura *Add query*, come mostrato in figura, per default il sistema assegna un nome generico con numerazione progressiva alla query, evidenziato in rosso nell'elenco *Query Name*, eventualmente modificabile. A questo punto, tramite il bottone di *Edit Query* si accede all'interfaccia *SQL Query Designer* per lo sviluppo dell'interrogazione, evidenziata nel riquadro rosso in Figura 6.6. Il dettaglio mostrato nella seguente Figura 6.7, rappresenta l'ambiente di sviluppo delle query SQL nella modalità grafica, detta *designer*.

Esso è costituito da tre aree principali. La prima, evidenziata nel riquadro blu, mostra l'elenco delle tabelle recuperate dalla sorgenti dati, in questo esempio si fa sempre riferimento al data warehouse discusso nei Capitoli precedenti, pertanto l'elenco riporta l'insieme delle tabelle dimensionali e dei fatti che ne fanno parte. Le tabelle che mantengono i dati interessanti che dovranno essere visualizzati sul report devono essere trascinate sull'area di lavoro evidenziata in verde. Automaticamente lo strumento mostra una

rappresentazione grafica per nome degli attributi che compongono la tabella che si possono scegliere, a seconda delle necessità, semplicemente spuntando il box adiacente a ciascun attributo.

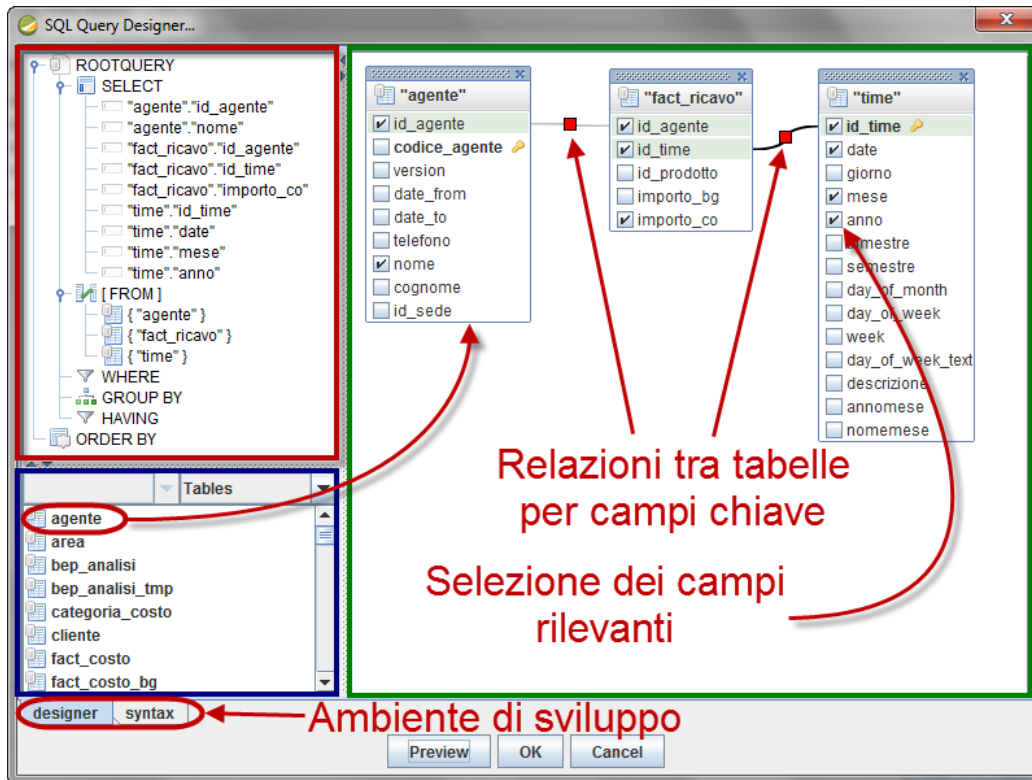


Figura 6.7: Ambiente grafico di sviluppo query SQL

Ad esempio, nella Figura 6.7 si scelgono le seguenti tuple per le rispettive tabelle: agente(id_agente, nome), time(id.time, data, mese, anno) e fact_ricavo(id_agente, id_time, importo_co). Una volta selezionati tutti i campi i cui valori sono ritenuti interessanti per la rappresentazione nel report, si passa all'unione delle tabelle inserite tramite opportuni campi chiave. Le relazioni sono evidenziate in figura e disposte graficamente in modo adiacente ai campi che definiscono l'unione (ad esempio le tabelle agente e fact_ricavo sono unite tramite il campo chiave id_agente). Anche in questo caso la giunzione avviene tramite la sovrapposizione dei due attributi con la tecnica del trascinarsi.

Come già osservato precedentemente, ad ogni azione effettuata dall'utente sull'interfaccia grafica corrisponde un'azione automatica dello strumento per la costruzione della query. Queste sequenze sono riportate nel riquadro evi-

denziato in rosso nella Figura 6.7, di cui se ne riporta il dettaglio in Figura 6.8.

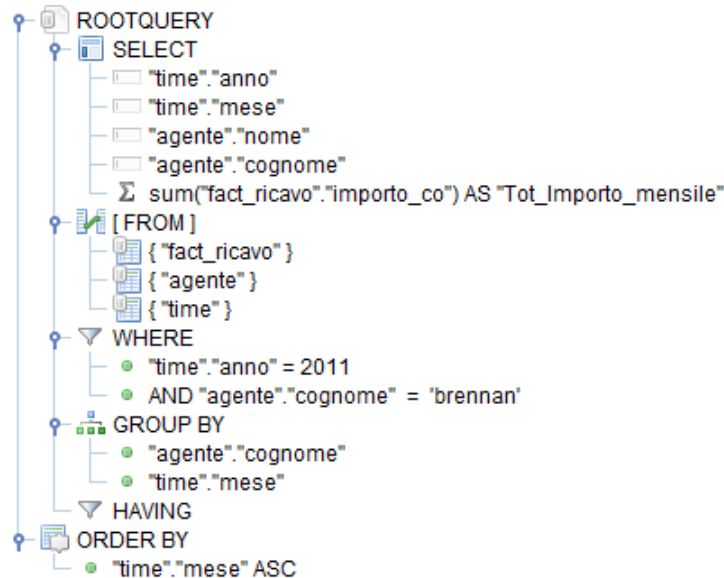


Figura 6.8: Esempio di Root Query

Il sistema propone la classica struttura ad albero a partire da una radice denominata *Rootquery*. Al suo interno sono già predefiniti i classici operatori SELECT, FROM, WHERE, GROUP BY, HAVING e ORDER BY. Mentre i comandi SELECT e FROM vengono parametrizzati automaticamente dal sistema in corrispondenza delle scelte utente, i parametri dei restanti WHERE, GROUP BY, HAVING e ORDER BY devono essere inseriti direttamente da quest'ultimo sfruttando le funzionalità opzionali sulla struttura della Rootquery. In altri termini, una volta selezionato un campo riportato nella clausola SELECT (ad esempio il campo agente.cognome), si può decidere se riportare lo stesso campo sulla GROUP BY o sull'ORDER BY tramite le opzioni *add to group-by* o *add to order-by*. Analogamente per i filtri sul campo WHERE, ma in questo caso l'opzione, eseguita direttamente sulla clausola, prevede una *add condition* con la quale si richiede di specificare l'attributo e il corrispettivo valore (come riportato nell'esempio si eseguono due condizioni del tipo `time.anno = 2011 AND agente.cognome = Brennan`). La clausola HAVING viene trattata esattamente nelle stesse modalità della WHERE. Si possono infine aggiungere funzioni e sotto query tramite le opzioni *add ex-*

pression e *add subquery* direttamente nella clausola SELECT. Nell'esempio la funzione utilizzata (SUM(fact_ricavo.ricovo_co) AS Tot_importo_mensile) definisce la somma totale dell'importo consuntivo dell'agente distinto sui vari mesi dell'anno. Nella clausola FROM invece è permesso il solo inserimento di una nuova sotto query sempre tramite l'opzione *add subquery*.

Lo strumento SQL Query Designer, oltre ad utilizzare una modalità grafica, si avvale anche della classica scrittura delle interrogazioni secondo la sintassi dell'SQL. Il bottone *Syntax* mostrato in Figura 6.9 riporta la sintassi dell'interrogazione sviluppata in precedenza con lo strumento grafico.

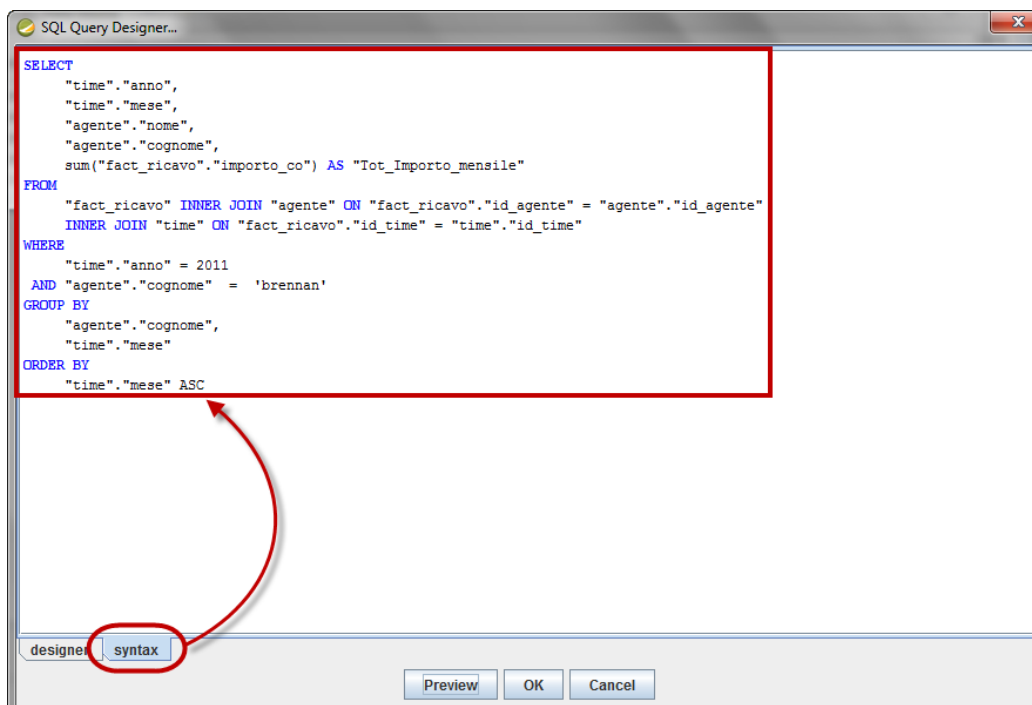


Figura 6.9: Esempio di query Syntax

A questo punto sarà cura dell'utente utilizzare la modalità grafica o alternativamente scrivere direttamente la query che recupera i dati di interesse. Fintanto che l'interrogazione viene sviluppata in modalità grafica, lo strumento propone automaticamente anche la sintassi corrispondente, mentre se si desidera scrivere direttamente la query, lo strumento non è in grado di definire la corrispettiva rappresentazione grafica di ciò che è stato realizzato.

Per concludere la fase di sviluppo delle query, al fine di ottenere un riscontro sulla correttezza dell'interrogazione stessa, lo strumento fornisce il

bottono *Preview* che mostra un'anteprima dei risultati corrispondenti. In Figura 6.10 è mostrato l'esempio dell'anteprima dei risultati ottenuti dall'esecuzione della query precedentemente sviluppata.

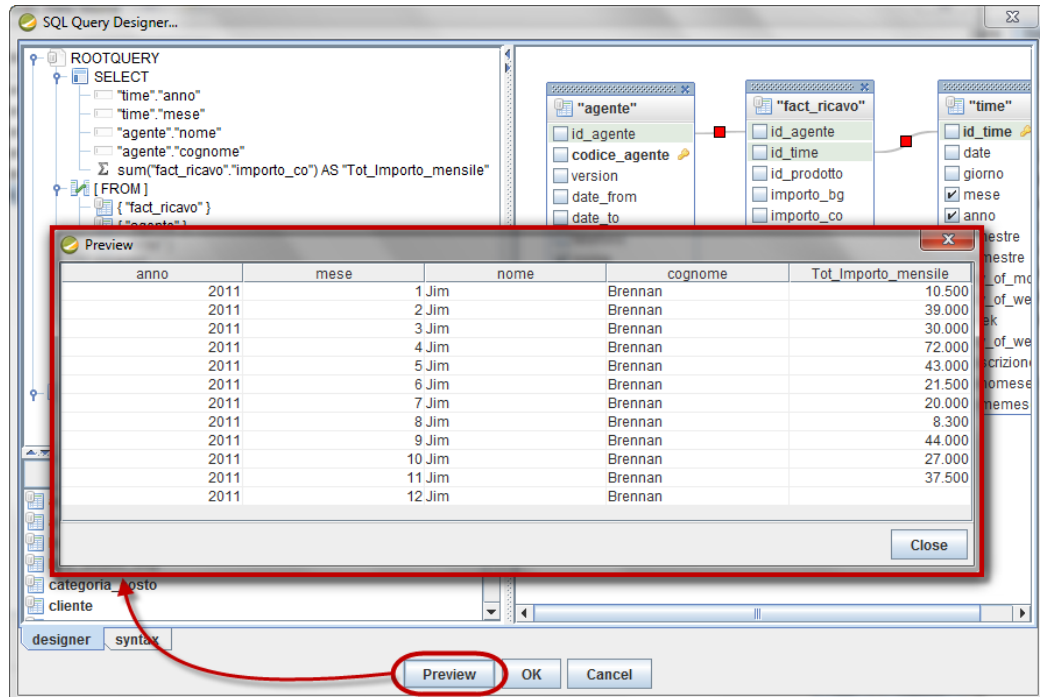


Figura 6.10: Anteprima sui risultati restituiti dalla query

Stabilita la connessione alla base di dati (o ad altre sorgenti) e una volta definita l'interrogazione dalla cui esecuzione si recuperano i dati ritenuti utili per l'informazione che si desidera fornire, occorre definire una struttura grafica di rappresentazione dell'informazione stessa. Per eseguire quest'ultimo passo ci si avvale dello strumento Report Designer mostrato in Figura 6.1 secondo le modalità di utilizzo già presentate.

In Figura 6.11 è mostrata la modalità di sviluppo grafico del report. Tramite la tecnica del trascinamento si può utilizzare l'elemento *label*, riportato sulla struttura mostrata al centro della figura, per definire il titolo del rapporto ed eventuali porzioni di testo statico che rimangono fisse durante la visualizzazione (questa procedura è evidenziata dalle frecce di colore rosso). Analogamente si utilizzano gli elementi *text-field* e *number-field* per visualizzare i contenuti recuperati dall'esecuzione di una o più query sulla base di dati (nell'esempio riportato gli elementi indicati sono evidenziati dalle frecce

di colore blu). Ogni volta che viene inserito un campo numerico o testuale ricevuto come risultato di una interrogazione, lo strumento richiede obbligatoriamente la definizione di quale attributo, fra tutti quelli restituiti dalla query, associare al campo stesso. Nei riquadri evidenziati in verde si nota la corrispondenza tra i risultati ottenuti tramite l'esecuzione della query e i campi di scelta nel menù a tendina per un elemento numerico inserito nel report.

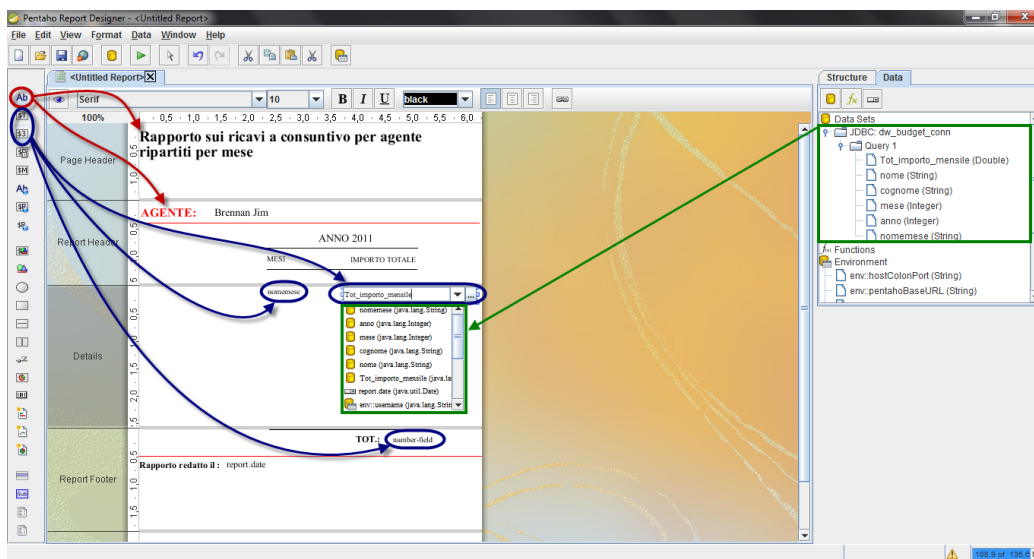


Figura 6.11: Esempio di costruzione del report con Report Designer

Nell'esempio di Figura 6.11 vengono rappresentati, nell'ordine: l'anno di riferimento, il nome e cognome dell'agente oggetto del rapporto, i nomi dei mesi dell'anno, gli importi totali dei ricavi preventivati a budget e a consuntivo ripartiti su ciascun mese e l'importo complessivo dei ricavi nell'intero anno, come informazioni recuperate dall'interrogazione il cui sviluppo è stato descritto in precedenza.

A termine dello sviluppo grafico del report, Report Designer permette di poter osservare un'anteprima dello stesso, utilizzando la funzione *Run* rintracciabile nella barra dei menù in testa sull'interfaccia, con la possibilità di scegliere molteplici formati (ad esempio *Print preview*, *PDF*, *HTML* e molti altri). L'anteprima del report di cui si sono riportati i passi progettuali è mostrata in Figura 6.12.

Come già descritto nei Capitoli precedenti in cui si trattano le modalità di utilizzo delle componenti di Pentaho C.E., anche in questo caso, per rendere il report progettato condivisibile agli utenti che utilizzano il sistema, si deve necessariamente salvare il file corrispondente (ad esempio *nome_report.prpt*) nel deposito file denominato repository.

Rapporto sui ricavi a consuntivo per agente ripartiti per mese		
AGENTE: Brennan Jim		
ANNO 2011		
MESI	TOTALE BUDGET	TOTALE CONSUNTIVO
Gennaio	25.800	10.500
Febbraio	25.800	39.000
Marzo	25.800	30.000
Aprile	25.800	72.000
Maggio	25.800	43.000
Giugno	25.800	21.500
Luglio	25.800	20.000
Agosto	25.800	8.300
Settembre	25.800	44.000
Ottobre	25.800	27.000
Novembre	25.800	37.500
Dicembre	25.800	
TOTALE:	309.600	352.800
	MARGINE:	+ 43.200

Rapporto redatto il : Fri Nov 25 17:07:46 CET 2011

Figura 6.12: Anteprima del report

Per fare questo ci si avvale della tecnica di pubblicazione del report. Ciascun utente potrà, a questo punto, richiamare il report direttamente dall'interfaccia user console di Pentaho, come mostrato in Figura 6.13.

Da quanto detto, si conclude che lo strumento Report Designer presenta alcuni punti di debolezza rispetto ad altri strumenti per la business intelligence. Ad esempio, può risultare difficoltoso, se non totalmente inefficiente, il suo utilizzo nel caso di interrogazioni relativamente complesse. Non è consentito

l'utilizzo di due o più data mart per generare nuovi campi calcolati infatti gli elementi che visualizzano le informazioni dei campi derivanti dall'esecuzione delle interrogazioni fanno riferimento ad un'unica query.

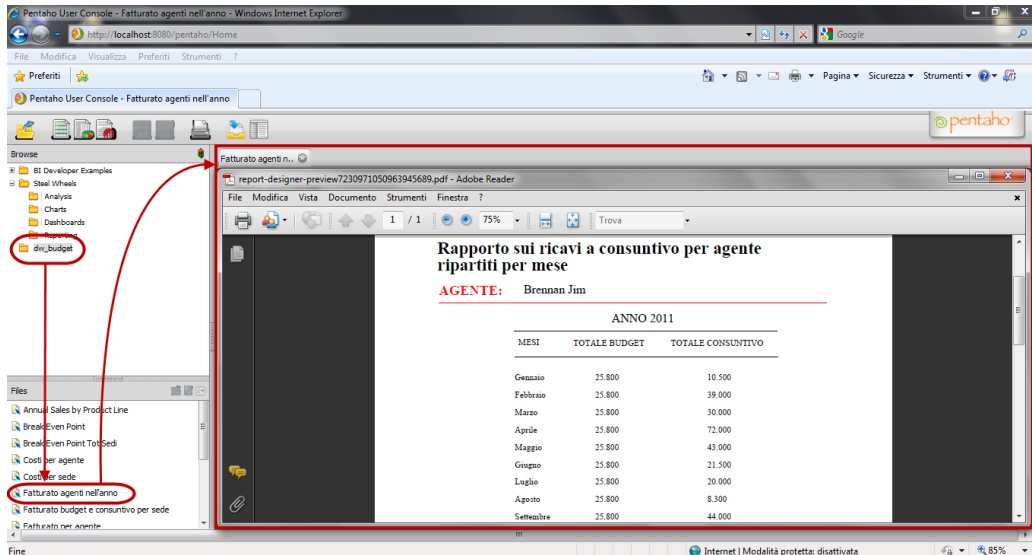


Figura 6.13: Visualizzazione del report dalla user console

File Xaction

Per ovviare a queste problematiche, Pentaho C.E. supporta comunque delle tecniche alternative o integrative di sviluppo della reportistica. Una di queste, ampiamente utilizzata per questo progetto è la tecnica delle sequenze d'azione, più comunemente denominata Xaction. Nel complesso questa tecnica si prefigge lo scopo di portare la generazione della reportistica ad un livello ampiamente superiore rispetto al Report Designer, sia nel dettaglio grafico che in termini di risultati ottenibili, migliorando la navigabilità e la consultazione del rapporto da parte dell'utente. Possiamo affermare che questa tecnica si avvicina molto alla visualizzazione nota come *Dashboard*, anche se non lo è a tutti gli effetti in quanto questa è prevista nella suite solamente nella versione *Pentaho Enterprise Edition* e non nella *Pentaho Community Edition* utilizzata per questo progetto.

Avvalersi della tecnica dell'Xaction significa sviluppare opportune procedure sequenziali che vanno dal recupero dei dati tramite interrogazioni SQL

da una sorgente fino alla visualizzazione degli stessi in modalità grafica, il tutto esprimibile in linguaggio XML. La rappresentazione finale del grafico, le caratteristiche di dinamicità e manipolabilità dello stesso, sono poi garantite anche dall'utilizzo di opportune librerie Flash e Javascript depositate su predefinite directories di sistema e richiamate nella sequenza d'azione.

Tipicamente un file Xaction è organizzato in una serie di azioni, eseguite nell'esatto ordine in cui queste si collocano, e tra le varie particolarità che le distinguono, risalta soprattutto il fatto che i dati e le informazioni trattate possono essere passati da un'azione alla successiva (purchè appartenenti sempre alla stessa sequenza) tramite l'utilizzo delle classiche strutture dati sfruttate nella programmazione (ad esempio tramite *buffer* dedicati o *array*).

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <action-sequence>
3. <title>Break Even Point</title>
4. <version>1</version>
5. <logging-level>DEBUG</logging-level>
6. <documentation>
7. <author>Marco Luchini - Matteo Pardini</author>
8. <description>Line chart</description>
9. <help/>
10. <result-type>Graph</result-type>
11. </documentation>
12.
13. <!-- Azione 1: impostazioni variabili utente -->
14.
15. <actions>
16. <inputs>
17. <anno type="string">
18. <default-value><![CDATA[2011]]></default-value>
19. </anno>
20. <id_sede type="string">
21. <default-value><![CDATA[1]]></default-value>
22. </id_sede>
23. </inputs>
24.
25. <outputs>
26. <ris type="string">
27. <destinations>
28. <response>content</response>
29. </destinations>
30. </ris>
31. </outputs>
32.
33. </resources/>
```

Figura 6.14: Sviluppo prima sequenza d'azione

Nell'esempio riportato si utilizzano quattro azioni fondamentali identificabili dai tag `< actions > ... < /actions >` comprese in un'unica sequenza d'azione richiamata tramite l'ausilio del tag `< action - sequence > ... < /action - sequence >`.

In Figura 6.14 è mostrata la prima delle quattro azioni utilizzata per il recupero dei dati di scelta dell'utente. Il tag `< inputs >` (linea 16) evidenzia le informazioni che devono essere ricevute tramite interfaccia utente per poter proseguire nella sequenza d'azione. In questo caso, trattandosi dell'esempio di generazione del grafico per la determinazione del punto di pareggio, si richiedono il codice identificativo della sede (*id_sede* alla linea 20) e l'anno di riferimento (*anno* alla linea 17). I possibili valori appartenenti alle due variabili in questione sono riportati su opportuni menù *text-box* sulla user console. I parametri contenuti nelle linee dalla 1 alla 11 forniscono informazioni aggiuntive quali: il tipo di documento e la codifica da utilizzare, il titolo del report che verrà mostrato all'utente, gli autori del grafico, una breve descrizione sul contenuto del rapporto e una descrizione sintetica sul tipo di visualizzazione utilizzato. Queste informazioni vengono riportate per completezza anche se ai fini della corretta visualizzazione del report non sono necessarie.

Terminata la prima azione si passa alla successiva, mostrata in Figura 6.15, caratterizzata dall'implementazione delle interrogazioni che il sistema dovrà eseguire sulla base di dati per il recupero delle informazioni utili da riportare sul report grafico.

Per la corretta esecuzione delle query occorre prima specificare il nome della connessione che il sistema dovrà utilizzare sfruttando i tag `< jndi > ... < /jndi >`, come mostrato nelle linee 44 e 64. Anche se la connessione specificata è la stessa per tutte le query che si desidera eseguire (in questo esempio si sfrutta la connessione denominata *conn_dw_budget*) occorre comunque specificarla prima di ciascuna interrogazione dal momento che il sistema non mantiene memorizzata questo tipo di informazione.

La particolarità dell'utilizzo di questa metodologia, rispetto al classico strumento Report Designer, come già osservato in precedenza, risiede proprio nel fatto di poter gestire dati restituiti dall'esecuzione di molteplici interrogazioni. In questo esempio, si eseguono due query, la prima dichiarata alla

linea 45, la seconda alla linea 65. La prima interrogazione è stata definita allo scopo di recuperare dall'apposita tabella nella base di dati tutti i record riferiti alla sede e all'anno scelti dall'utente nella fase precedente.

```

34. <!-- Azione 2: Connessione e esecuzione query sul data warehouse -->
35.
36. <actions>
37.   <action-definition>
38.     <component-name>SQLLookupRule</component-name>
39.     <action-type>QueryDaData</action-type>
40.     <action-outputs>
41.       <query-result type="result-set" mapping="datiGrafico"/>
42.     </action-outputs>
43.     <component-definition>
44.       <jndi><![CDATA[conn_dw_budget]]></jndi>
45.       <query><![CDATA[SELECT
46.                   DATE_FORMAT(b.date, '%d-%m') AS data,
47.                   b.cf, b.cv, b.ct, b.rt, b.anno, s.nome
48.               FROM
49.                   bep_analisi b, sede s
50.               WHERE
51.                   b.date is not null and b.anno=2011 and
52.                   b.id_sede=1 and s.id_sede=b.id_sede]]>
53.     </query>
54.   </component-definition>
55. </action-definition>
56.
57.   <action-definition>
58.     <component-name>SQLLookupRule</component-name>
59.     <action-type>QueryDaData</action-type>
60.     <action-outputs>
61.       <query-result type="result-set" mapping="datiGrafico2"/>
62.     </action-outputs>
63.     <component-definition>
64.       <jndi><![CDATA[conn_dw_budget]]></jndi>
65.       <query><![CDATA[SELECT
66.                   DATE_FORMAT(b.date, '%d-%m-%Y') AS data
67.               FROM
68.                   bep_analisi b
69.               WHERE
70.                   b.date is not null and b.anno=2011 and
71.                   b.id_sede=1 and b.rt >= b.ct
72.               ORDER BY
73.                   b.date LIMIT 1]]>
74.     </query>
75.   </component-definition>
76. </action-definition>

```

Figura 6.15: Sviluppo seconda sequenza d'azione

Le informazioni estratte per ciascun record si riferiscono in particolar modo al valore del costo fisso, costo variabile, costo totale e ricavo totale, tutti dati utili per lo sviluppo del grafico sul piano cartesiano. La seconda query è invece riservata per l'estrazione di un'unico record dalla medesima tabella. Il

record in questione sarà quello per cui vale l'equazione $b.rt \geq b.ct$, in altri termini il record in cui si riscontra il superamento o l'uguaglianza del valore dei ricavi totali aziendali rispetto ai costi totali (linea 71). L'informazione interessante che verrà estratta sarà quindi la data corrispondente, ovvero la data prevista per il raggiungimento del punto di pareggio.¹

Per concludere, nelle linee 41 e 61 si definiscono il tipo e il nome delle strutture dati che dovranno mantenere le informazioni restituite, dichiarate poi nella fase successiva, rispettivamente per la prima e la seconda query. In questo caso il tipo della struttura dati è un *result-set* e il nome *datiGrafico*.

Successivamente, in Figura 6.16 è mostrata la terza fase dell'intera sequenza d'azione che riguarda il salvataggio vero e proprio dei dati restituiti sui *result-set*. Allo scopo occorre prima dichiarare quali saranno i dati di input (linea 83), in questo caso i due *result-set*, poi dichiarare il tipo di output degli stessi (linea 87).

Una volta fatte le dovute assunzioni si procede all'inserimento fisico delle informazioni nella struttura dati tramite l'ausilio di programmazione Java, compreso nel tag `< script > ... < /script >` (mostrato nelle linee dalla 98 alla 124). Quasi tutto lo script si basa sull'utilizzo dei metodi `.getValueAt()` e `.getRowCount()` rispettivamente per l'estrazione fisica del dato e il conteggio delle righe (o record) restituite. Con il primo metodo si vanno a salvare i dati riferiti a ciascun record (i valori di costo fisso, costo variabile, costo totale e ricavo totale) sulla struttura dati riservata, con il secondo si recupera il numero totale di record restituiti utilizzato come parametro di controllo nel costrutto *for* di linea 113.

La variabile *out_str* dichiarata nella linea 106 è impostata inizialmente al valore di esito negativo per il raggiungimento del punto di pareggio. Solo dopo aver eseguito il controllo con il costrutto *if* alla linea 109, con cui si riscontra l'effettivo raggiungimento dell'obiettivo, la variabile può essere sovrascritta con una nuova stringa riportante l'esito positivo e la data stimata. Sarà poi compito della fase successiva stampare a video il valore contenuto su questa variabile. Il controllo in questione risulta piuttosto semplice, in

¹Si rimanda il lettore alla consultazione delle Sezioni 2.3, 3.8 e 4.2.2 per l'osservazione delle metodologie, tecniche e formule implementate per la determinazione del punto di pareggio.

quanto occorre solamente verificare l'esistenza, nel corrispondente result-set, dell'eventuale unico record restituito dall'esecuzione della seconda query mostrata in precedenza.

```

77. <!-- Azione 3: salvataggio dei dati sui result-set -->
78.
79. <actions>
80. <action-definition>
81.   <component-name>JavascriptRule</component-name>
82.   <action-type>JavaScript</action-type>
83.   <action-inputs>
84.     <datiGrafico type="result-set"/>
85.     <datiGrafico2 type="result-set"/>
86.   </action-inputs>
87.   <action-outputs>
88.     <dates type="string"/>
89.     <cf type="string"/>
90.     <cv type="string"/>
91.     <ct type="string"/>
92.     <rt type="string"/>
93.     <ds type="string"/>
94.     <ris type="string"/>
95.     <out_str type="string"/>
96.   </action-outputs>
97.   <component-definition>
98.     <script><![CDATA[var count=datiGrafico.getRowCount();
99.         var dates="";
100.        var cf="";
101.        var cv="";
102.        var ct="";
103.        var rt="";
104.        var ds="";
105.        var ris="<p align='center'> nodata </p>";
106.        var out_str="Il punto di pareggio non e' atteso per la sede di " +
107.            datiGrafico.getValueAt(0,6) + " nell'anno " +
108.            datiGrafico.getValueAt(0,5);
109.        if (datiGrafico2.getRowCount()=="1")
110.            {out_str="Data stimata per la sede di " + datiGrafico.getValueAt(0,6) +
111.                "nell'anno" + datiGrafico.getValueAt(0,5) + ":" +
112.                datiGrafico2.getValueAt(0,0)};
113.        for(var i=0;i<count;i++)
114.            {dates+="<value xid='"+i+"'>" +datiGrafico.getValueAt(i,0)+"</value>";
115.            cf+="<value xid='"+i+"'>" +datiGrafico.getValueAt(i,1)+"</value>";
116.            cv+="<value xid='"+i+"'>" +datiGrafico.getValueAt(i,2)+"</value>";
117.            ct+="<value xid='"+i+"'>" +datiGrafico.getValueAt(i,3)+"</value>";
118.            rt+="<value xid='"+i+"'>" +datiGrafico.getValueAt(i,4)+"</value>";}
119.
120.        ds="<chart><series>" +dates+"</series><graphs>
121.            <graph gid='0'>" +cf+"</graph><graph gid='1'>" +cv+"</graph>
122.            <graph gid='2'>" +ct+"</graph><graphgid='3'>" +rt+"
123.            </graph></graphs></chart>";
124.        ]]></script>
125.
126.   </component-definition>
127. </action-definition>

```

Figura 6.16: Sviluppo terza sequenza d'azione

Per concludere, la variabile *ds* dichiarata alla linea 120 ha come unico scopo

la corretta associazione tra i dati gestiti in questa fase nello script e le linee mostrate sul grafico, le cui impostazioni sono dichiarate nell'ultima azione della sequenza, il cui dettaglio è mostrato in Figura 6.17.

```

128. <!-- Azione 4: costruzione del grafico -->
129.
130. <actions>
131. <action-definition>
132.   <component-name>TemplateComponent</component-name>
133.   <action-type>Message Template</action-type>
134.   <action-inputs>
135.     <ds type="string"/>
136.     <out_str type="string"/>
137.   </action-inputs>
138.   <action-outputs>
139.     <output-message type="string" mapping="ris"/>
140.   </action-outputs>
141.
142. <component-definition>
143.   <template><![CDATA[<script type="text/javascript"
144.     src="/sw-style/amcharts/flash/swfobject.js">
145.     </script><div id="amcharts_1">
146.     You need to upgrade your Flash Player</div>
147.     <script type="text/javascript">
148.
149.     var params = {
150.       bgcolor:"#FFFFFF"
151.     };
152.
153.     var flashVars = {
154.       path: "/sw-style/amcharts/flash/",
155.       chart_settings: encodeURIComponent("<settings ></settings>"),
156.       chart_data: encodeURIComponent("{ds}")
157.     };
158.
159.     swfobject.embedSWF("/sw-style/amcharts/flash/amline.swf",
160.       "amcharts_1", "100%", "100%", "8.0.0",
161.       "/sw-style/amcharts/flash/expressInstall.swf",
162.       flashVars, params);
163.
164.     </script>]]>
165.   </template>
166.
167. </component-definition>
168. </action-definition>
169. </actions>
170. </action-sequence>

```

Figura 6.17: Sviluppo quarta sequenza d'azione

Dal momento in cui le variabili appena descritte (*out_str* e *ds*) vengono passate all'azione come parametri di input (linea 134), il sistema ha a disposizione tutti i parametri utili per la rappresentazione grafica. Le impostazioni per tale rappresentazione sono mantenute all'interno dello script compreso nel tag `< template > ... < /template >` mostrato nelle linee da 143 a 165. Per otte-

nera una corretta visualizzazione del grafico occorre in primo luogo definire all'interno dello script i *path-name* per il recupero delle librerie Javascript e Flash, definiti rispettivamente nelle variabili *src* alla linea 144 e *path* alla linea 154. Un'altra impostazione utile, ma non obbligatoria, è la dichiarazione del colore di sfondo mostrato durante la visualizzazione. In questo esempio si utilizza uno sfondo bianco, mostrato alla linea 149, per risaltare meglio il grafico stesso. Al contrario sono indispensabili i parametri *chart_settings* e *chart_data* riportati nelle linee 155 e 156. Essi costituiscono la parametrizzazione sia in termini di stile grafico che in termini di informazioni contenute per il report grafico. L'istruzione *encodeURIComponent* ha lo scopo di assegnare la corretta codifica delle variabili inserite, che nel caso dei dati si riferisce alla variabile *ds* dichiarata precedentemente, mentre nel caso dello stile di rappresentazione del grafico si riferisce all'intero codice sorgente XML compreso nei tag `< settings > ... < /settings >`.²

Esistono molteplici tecniche per recuperare il codice XML sorgente per le rappresentazioni di grafici, siano essi tipiche raffigurazioni a barre, a torta, a linee ecc.

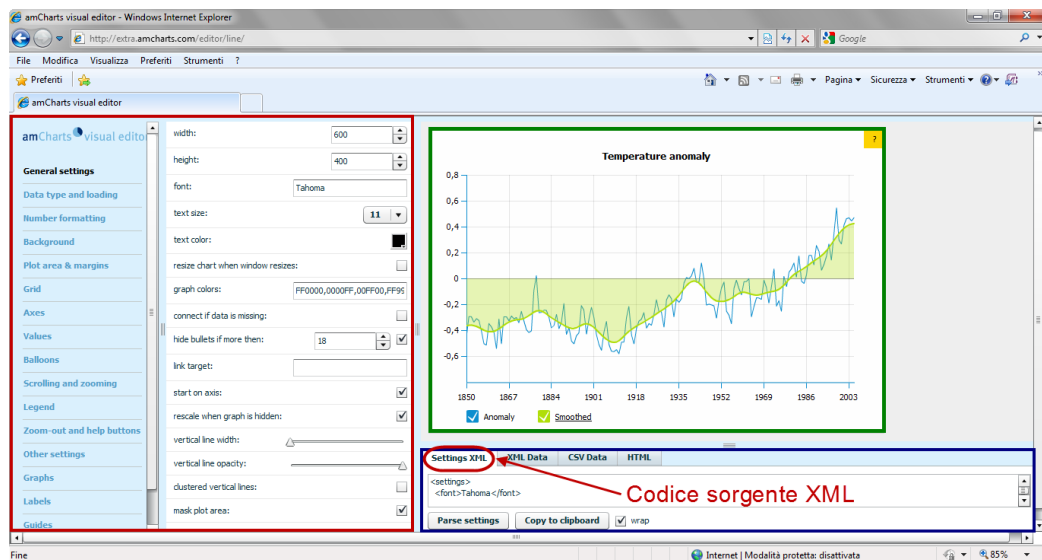


Figura 6.18: Interfaccia utente della risorsa AmCharts

²In questo esempio il codice XML non viene mostrato perchè ritenuto irrilevante ai fini della comprensione della metodologia di progettazione di report grafici mediante sequenza d'azione.

Quella utilizzata in questo progetto (per tutti i report grafici progettati), sfrutta la risorsa *AmCharts* facilmente utilizzabile via web. In Figura 6.18 è mostrata l'interfaccia utente dello strumento una volta scelto il tipo di grafico che si vuole creare.

Essa si distingue su tre aree fondamentali, la prima evidenziata nel riquadro rosso, riporta tutti gli elementi del grafico parametrizzabili dall'utente. Tra questi si trovano elementi di carattere generale (*General settings*), elementi di impostazione dello sfondo (*Background*), elementi di configurazione degli assi cartesiani (*Axes*), della griglia (*Grid*), dei valori (*Values*), della legenda (*Legend*) ecc.

In concomitanza ad ogni modifica dei parametri riportati nell'elenco mostrato nell'area evidenziata in rosso, lo strumento genera automaticamente la corrispondente visualizzazione del grafico mostrata nel riquadro evidenziato in verde. Una volta definite correttamente tutte le impostazioni dei parametri secondo le esigenze dell'utente, nel riquadro evidenziato in blu nella figura, si recupera il codice sorgente XML che mantiene tutte le impostazioni corrispondenti. A questo punto il codice può essere copiato ed inserito nella sequenza d'azione all'interno dei tag `< settings > ... < /settings >`, come mostrato in Figura 6.17.

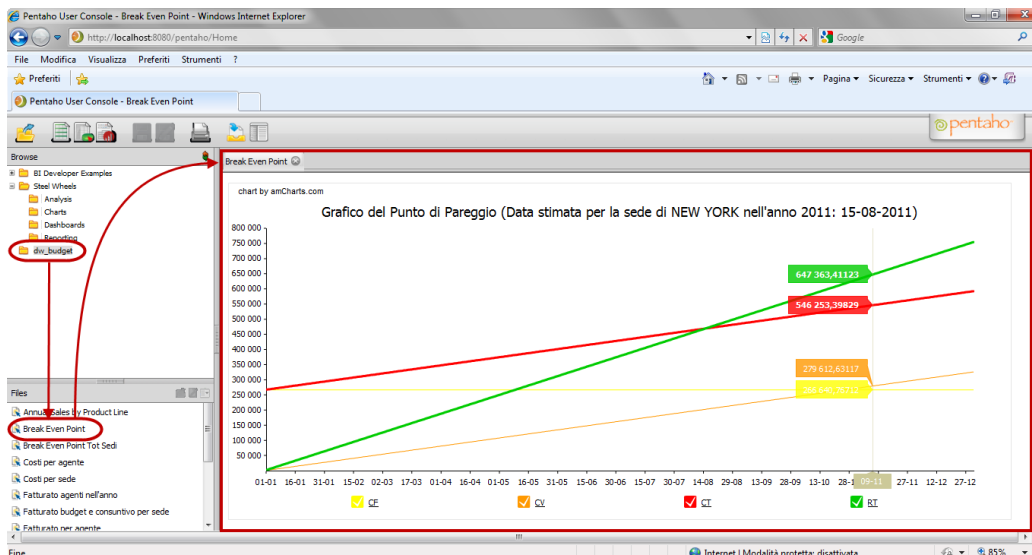


Figura 6.19: Visualizzazione del grafico dalla user console

Al fine di rendere condivisibile il rapporto grafico, le cui modalità di progetta-

zionate sono state appena discusse, il corrispondente file (ad esempio *nome_report.xaction*) deve necessariamente essere salvato all'interno della directory repository (come già osservato per altri tipi di file generati dall'utilizzo degli strumenti di BI di Pentaho C.E.). L'utente potrà lanciare l'esecuzione del file direttamente dalla user console di Pentaho e consultare il grafico, come mostrato in Figura 6.19.

6.2 Rapporti basati sui requisiti di analisi

Secondo quanto detto a conclusione del Capitolo precedente, le informazioni sui costi e ricavi aziendali, le cui analisi sono rese possibili grazie alla progettazione dei rispettivi cubi OLAP, possono essere presentate sottoforma di report tabellari, intuitivi e semplici da analizzare per l'utente finale. Si mostrano quindi alcuni esempi di tali report, generati sfruttando lo strumento Saiku, basati sui requisiti di analisi e discussi in sede di progettazione del data warehouse nel Capitolo 3.

Rapporti per l'analisi dei dati di costo

Sulla base dei requisiti di analisi numero 2 e 8 si genera il report mostrato in Figura 6.20.

2	Totale importo a budget delle voci di costo per categoria e per tipologia
8	Totale importo a consuntivo delle voci di costo per categoria e per tipologia

In questo caso interessano gli importi complessivi specifici di ciascuna voce di costo, distinta per tipologia, nell'anno in corso per tutte le sedi. La dimensione Sede scompare dall'output finale e ciò lascia intendere all'utente che i dati riportati siano complessivi per tutte le sedi di riferimento.

tipo	nome	budget_c	consuntivo_c	scostamento_c
costi fissi diretti	Stipendio lordo base	130.900	113.450	-17.450
costi fissi indiretti	Affitto sede	375.600	344.300	-31.300
	Pulizia ufficio	67.850	62.880	-4.970
	Canone Locazione arredamento	189.600	177.000	-12.600
	Esazione CCIAA	85.200	74.510	-10.690
	Consulenti amministrativi e del lavoro	196.800	148.310	-48.490
	Canone tenuta conto CRF	18.000	16.570	-1.430
costi variabili diretti	Auto noleggio	25.550	21.920	-3.630
	Cellulare	9.930	8.725	-1.205
	Rimborso pie di lista	25.750	22.015	-3.735
costi variabili indiretti	Cancelleria	44.400	36.850	-7.550
	Omaggi a clienti	52.200	35.110	-17.090
	Costo azienda base	136.000	100.900	-35.100
	Energia elettrica	117.600	110.200	-7.400
	Noleggio autovetture	171.600	163.680	-7.920
	Quote associative	336.000	307.800	-28.200
	Vidimazione libri sociali	229.200	186.580	-42.620
	Commissioni bancarie CRF	82.800	65.510	-17.290
	Imposta di bollo	24.000	16.860	-7.140
	Corsi formativi	98.100	84.800	-13.300

Figura 6.20: Report per i requisiti 2 e 8

Rapporti per l'analisi dei dati di ricavo

Sulla base dei requisiti di analisi numero 6 e 12 si genera il report mostrato in Figura 6.21.

6	Ricavo a budget per sede, per l'anno corrente
12	Importo del fatturato totale per sede, per l'anno corrente

Il fatturato totale di una singola sede, sia nella misura a budget che in quella a consuntivo, è rappresentato per gli agenti che fanno parte della sede stessa. Fintanto che una sede ha un solo agente associato (ad esempio la sede di Milano), le due misure corrisponderanno al complessivo fatturato aziendale. Qualora una sede presenti più agenti associati (ad esempio la sede di New York), allora il suo fatturato complessivo, espresso con le misure riportate, deve essere considerato come la somma dei valori indicati per ogni suo agente riferito.

anno	sede	cognome	budget_r	consuntivo_r	scostamento_r
2011	Madrid	Olliver	456.000	516.100	60.100
	Milano	Westin	432.000	490.100	58.100
	Monaco	Jensen	408.000	470.600	62.600
	New York	Brennan	309.600	352.800	43.200
		Smith	324.000	348.300	24.300
	Toronto	Bronsen	504.000	430.900	-73.100

Figura 6.21: Report per i requisiti 6 e 12

Rapporti per l'analisi complessiva dei dati

Si conclude la rassegna degli esempi di report basati sui requisiti di analisi presentando l'output derivante dalla progettazione del cubo virtuale discusso nel Capitolo precedente. L'output di Figura 6.22, strutturato secondo richiesta del committente, si discosta leggermente dalla visualizzazione grafica del cubo virtuale di Figura 5.19.

cognome	2011					
	budget_c	budget_r	consuntivo_c	consuntivo_r	Utile budget	Utile consuntivo
Brennan	329.600	309.600	271.780	352.800	-20.000	81.020
Bronsen	457.690	504.000	413.540	430.900	48.310	17.380
Jensen	365.840	408.000	335.060	470.600	42.160	135.540
Olliver	419.000	456.000	371.770	516.100	37.000	144.330
Smith	322.310	324.000	285.675	348.300	1.690	82.625
Westin	388.640	432.000	339.245	490.100	45.380	150.855

Figura 6.22: Report dell'utile consuntivato derivante dalle vendite per agenti

L'attenzione è rivolta soprattutto alla valutazione dell'utile complessivo pianificato a budget e realizzato a consuntivo in base agli importi dei ricavi e dei costi registrati per ciascun agente di vendita nell'anno considerato. Si tralasciano pertanto gli altri attributi della dimensione Data, quali il semestre, il trimestre e il mese di riferimento.

Si ricorda che i rapporti appena mostrati sono creati con l'ausilio dello strumento Saiku, ampiamente discusso nella Sezione 5.2.1. Pertanto, grazie alle caratteristiche di quest'ultimo, è lasciata all'utente finale la piena libertà

di creare il report che maggiormente si adatta alle sue esigenze, aggiungendo o togliendo dimensioni, attributi dimensionali e misure dalla visualizzazione.

6.3 Rapporti grafici per le analisi prestazionali

Usufruento della particolare tecnica che utilizza le sequenze d'azione, trattata nelle Sezioni precedenti, si mostrano ora gli output risultanti distinguendoli, a seconda della complessità della scrittura della query SQL all'interno del file Xaction, in tre categorie:

- Report con query *relativamente semplici*: si basano sui rapporti più comuni che richiedono il raggruppamento dei dati per il calcolo di metriche basate su somme di misure e cardinalità di insiemi.
- Report con query *relativamente complesse*: si basano sui rapporti che estendono quelli del caso precedente, richiedendo confronti fra più metriche calcolate.
- Report con query *complesse*: si basano sui rapporti che richiedono di estendere il risultato dei casi precedenti con altre metriche che comportano calcoli di altre su sottoinsiemi del risultato con funzioni di aggregazione non previste dall'SQL tradizionale.

Rapporti relativamente semplici

In questa prima tipologia rientrano tutte le interrogazioni eseguite per rispondere ai requisiti di analisi definiti in sede di progettazione in accordo con il committente (come specificato nel Capitolo 3). Si ritrovano i rapporti più comuni che richiedono al più filtri e raggruppamenti dei dati per il calcolo di metriche basate su somme di misure e cardinalità di insiemi. Di seguito si mostrano alcuni esempi tra i più significativi.

Nella Figura 6.23 vengono rappresentati, con l'ausilio di un grafico a torta, l'ammontare complessivo dei costi rilevati a consuntivo per ogni sede. La query SQL che ha permesso di estrapolare dal data warehouse i dati utili alla rappresentazione del grafico è quella di Figura 6.24.

Grafico sulla distribuzione dei costi tra le sedi aziendali

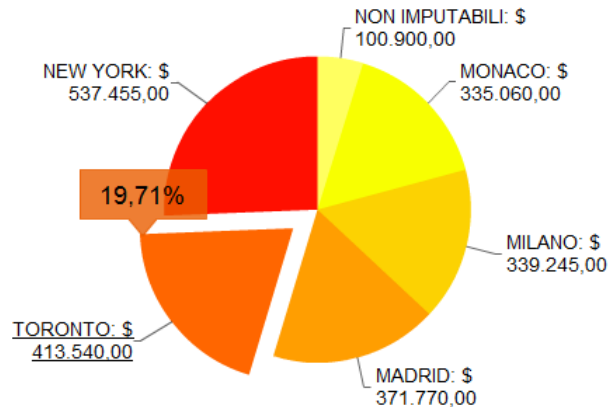


Figura 6.23: Grafico di ripartizione dei costi totali a consuntivo per sede

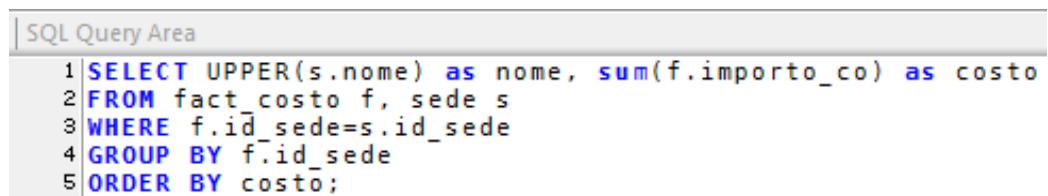


Figura 6.24: Query SQL per il grafico di Figura 6.23

Un ulteriore esempio, che gode di una rappresentazione grafica a barre, diversa dalla precedente, è mostrato in Figura 6.25. Esso riporta il fatturato nell'anno 2011 sia a budget che a consuntivo per ciascuna sede.

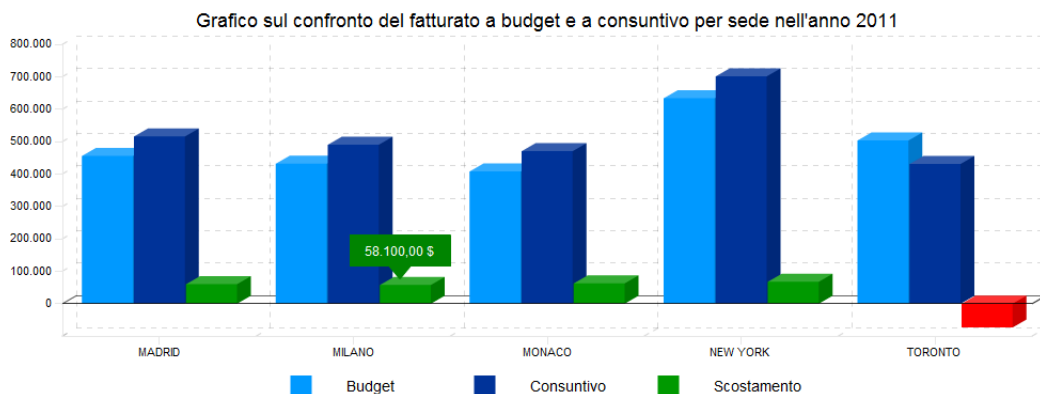


Figura 6.25: Grafico per la valutazione simultanea degli scostamenti per sede

La query SQL che ha permesso l'estrazione dei dati utili per la rappresentazione del grafico, è riportata in Figura 6.26.

```
SQL Query Area
1 SELECT s.nome, sum(importo_bg) as budget, sum(importo_co) as
2 consuntivo, sum(importo_co)-sum(importo_bg) as scostamento
3 FROM fact_ricavo f, agente a, sede s
4 WHERE f.id_agente=a.id_agente and a.id_sede=s.id_sede
5 GROUP BY s.id_sede
6 ORDER BY s.nome;
```

Figura 6.26: Query SQL per il grafico di Figura 6.25

La necessità di sviluppare questa rappresentazione deriva dalla volontà di poter osservare contemporaneamente i valori degli scostamenti di tutte le sedi, favorendone un confronto immediato e veloce.

Rapporti relativamente complessi

Nella seconda categoria rientrano le query relativamente complesse, ovvero quelle interrogazioni utili alla produzione di risultati sottoforma di rapporti per confronto o alternativamente di tabelle a doppia entrata, come ad esempio il confronto dei ricavi (in questo caso specifico anche dei costi) per agente, per prodotto e per mese su anni differenti. Per quanto concerne il progetto in oggetto, questa sezione di reportistica non è stata implementata, avendo a disposizione solo i dati riferiti all'anno attualmente in corso.³

Nonostante ciò, ipotizzando di essere in possesso degli adeguati dati per gli anni 2010 e 2011, si mostra in Figura 6.27 l'interrogazione SQL che permette il recupero dei dati di interesse dal data warehouse di Figura 3.7 al fine di fornire un confronto del fatturato dei singoli agenti di vendita nei vari mesi di entrambi gli anni riferiti. In Figura 6.28 si mostra invece l'output restituito da questa query, tenendo in considerazione il fatto che esso sia privo di attendibilità dal momento che i dati inerenti all'anno di riferimento 2010 sono stati generati appositamente allo scopo unico di verifica del corretto funzionamento dell'interrogazione stessa.

Al fine dell'analisi, è necessaria una giunzione esterna completa per recuperare tutti gli agenti che hanno prestato servizio nei due anni presi in considerazione, ivi compresi eventuali licenziamenti o nuovi assunti che quindi possono comparire in un anno ma non nell'altro. Purtroppo, la base di

³Questo argomento viene trattato nella successiva Sezione dedicata alle conclusioni, come spunto per gli sviluppi futuri del progetto.

dati progettata con MySQL non consente le giunzioni di tipo FULL OUTER JOIN tra due o più tabelle; pertanto, per ovviare a questo inconveniente, si rende necessario l'utilizzo di una UNION eseguita tra due tabelle. In particolare, la prima tabella è ottenuta tramite il comando LEFT OUTER JOIN tra le tabelle denominate A2011 e A2010 e analogamente la seconda tabella è generata tramite l'esecuzione del comando RIGHT OUTER JOIN sempre effettuato tra A2011 e A2010. Le due tabelle principali (A2011 e A2010) contengono rispettivamente il fatturato mensile di ogni agente per gli anni considerati, desunto a partire dalla tabella dei fatti mantenuta nel data warehouse inerente ai ricavi.

```

SQL Query Area
1 SELECT AGENTE,MESE, FATTURATO2011, FATTURATO2010,
2   CASE
3     WHEN ((FATTURATO2011 IS NULL) and (FATTURATO2010 IS NULL)) THEN 0
4     WHEN FATTURATO2011 IS NULL THEN -(FATTURATO2010)
5     WHEN FATTURATO2010 IS NULL THEN FATTURATO2011
6     ELSE FATTURATO2011-FATTURATO2010
7   END as DELTA,
8   CASE
9     WHEN ((FATTURATO2011 IS NULL) and (FATTURATO2010 IS NULL)) THEN 0
10    WHEN FATTURATO2011 IS NULL THEN -100
11    WHEN FATTURATO2010 IS NULL THEN 100
12    ELSE ROUND(100*(FATTURATO2011 - FATTURATO2010 ) /FATTURATO2011)
13  END as DELTA_PERC
14 FROM
15 (SELECT AGENTE, MESE, FATTURATO2011, FATTURATO2010, FATTURATO2011-FATTURATO2010 as DELTA,
16   ROUND(100*(FATTURATO2011 - FATTURATO2010 ) /FATTURATO2011) as DELTA_PERC
17 FROM
18 (SELECT CONCAT (a.cognome, ' ',a.nome) as AGENTE, t.nomemese as MESE,
19   sum(importo_co) as FATTURATO2011
20 FROM fact_ricavo f, time t, agente a
21 WHERE t.id_time=f.id_time and a.id_agente=f.id_agente and t.anno=2011
22 GROUP BY a.id_agente, t.annomese) as A2011 LEFT OUTER JOIN
23
24 (SELECT CONCAT (a.cognome, ' ',a.nome) as AGENTE, t.nomemese as MESE,
25   sum(importo_co) as FATTURATO2010
26 FROM fact_ricavo f, time t, agente a
27 WHERE t.id_time=f.id_time and a.id_agente=f.id_agente and t.anno=2010
28 GROUP BY a.id_agente, t.annomese) as A2010 USING (agente,mese)) as X
29
30 UNION
31
32 (SELECT AGENTE, MESE, FATTURATO2011, FATTURATO2010, FATTURATO2011-FATTURATO2010 as DELTA,
33   ROUND(100*(FATTURATO2011 - FATTURATO2010 ) /FATTURATO2011) as DELTA_PERC
34 FROM
35 (SELECT CONCAT (a.cognome, ' ',a.nome) as AGENTE, t.nomemese as MESE,
36   sum(importo_co) as FATTURATO2011
37 FROM fact_ricavo f, time t, agente a
38 WHERE t.id_time=f.id_time and a.id_agente=f.id_agente t.anno=2011
39 GROUP BY a.id_agente, t.annomese) as A2011 RIGHT OUTER JOIN
40
41 (SELECT CONCAT (a.cognome, ' ',a.nome) as AGENTE, t.nomemese as MESE,
42   sum(importo_co) as FATTURATO2010
43 FROM fact_ricavo f, time t, agente a
44 WHERE t.id_time=f.id_time and a.id_agente=f.id_agente and t.anno=2010
45 GROUP BY a.id_agente, t.annomese) as A2010 USING (agente,mese));

```

Figura 6.27: Query SQL di confronto del fatturato degli agenti su due anni

Un'ulteriore SELECT è stata introdotta allo scopo di selezionare i campi opportuni e calcolare la differenza tra gli importi nei due anni, restituendone anche la loro variazione percentuale. Per concludere, alla SELECT più esterna viene lasciato il compito di gestire alcune casistiche relativamente alla

possibile presenza di campi a valore nullo eventualmente presenti nelle due tabelle di origine.

AGENTE	MESE	FATTURATO2011	FATTURATO2010	DELTA	DELTA_PERC
Brennan Jim	Gennaio	10500	10500	0	0
Brennan Jim	Febbraio	39000	39000	0	0
Brennan Jim	Marzo	30000	30000	0	0
Brennan Jim	Aprile	72000	72000	0	0
Brennan Jim	Maggio	43000	43000	0	0
Brennan Jim	Giugno	21500	21500	0	0
Brennan Jim	Luglio	20000	20000	0	0
Brennan Jim	Agosto	8300	18300	-10000	-120
Brennan Jim	Settembre	44000	44000	0	0
Brennan Jim	Ottobre	27000	27000	0	0
Brennan Jim	Novembre	37500	37500	0	0
Brennan Jim	Dicembre	NULL	45000	-45000	-100
Smith Sarah	Gennaio	16400	20900	-4500	-27
Smith Sarah	Febbraio	38000	38000	0	0
Smith Sarah	Marzo	30300	30300	0	0
Smith Sarah	Aprile	35100	35100	0	0
Smith Sarah	Maggio	38500	38500	0	0
Smith Sarah	Giugno	35500	35500	0	0
Smith Sarah	Luglio	26700	44700	-18000	-67
Smith Sarah	Agosto	14500	19200	-4700	-32
Smith Sarah	Settembre	43000	43000	0	0
Smith Sarah	Ottobre	41800	41800	0	0
Smith Sarah	Novembre	28500	32700	-4200	-15
Smith Sarah	Dicembre	NULL	29900	-29900	-100

Figura 6.28: Output restituito dalla query di Figura 6.27

Rapporti complessi senza l'SQL analitico

La terza tipologia comprende le interrogazioni complesse che si possono realizzare senza l'SQL analitico. Col termine SQL analitico si intende tutto quell'insieme di estensioni, integrative dell'SQL classico, utili all'implementazione di particolari funzioni che altrimenti sarebbero difficili, se non in alcuni casi addirittura impossibili, da realizzare. Rientrano in questa tipologia, tra i vari esempi, i partizionamenti di insiemi di record, le funzioni di rango e l'utilizzo di funzioni di aggregazione su finestre.

Nel presente caso di studio, la necessità di utilizzare funzioni analitiche è sorta dal momento in cui il committente ha avanzato la richiesta di poter realizzare report raffiguranti, per ciascun mese dell'anno, i tre agenti ed i tre prodotti che hanno fornito il fatturato maggiore. Pertanto si sono realizzati rispettivamente i report seguenti mostrati nelle Figure 6.29 e 6.30.

Non disponendo MySQL di una funzione analitica per il calcolo della predetta specifica, nella fattispecie della funzione RANK(), abbiamo proceduto

sfruttando l'esecuzione della query presentata nella Figura 6.31.

Mese: Aprile	
Agente	Fatturato
Brennan Jim	72000.0
Olliver Chris	51600.0
Westin Will	50600.0

Mese: Maggio	
Agente	Fatturato
Bronsen Sally	63700.0
Westin Will	43100.0
Brennan Jim	43000.0

Mese: Giugno	
Agente	Fatturato
Olliver Chris	50700.0
Westin Will	48200.0
Jensen Max	46800.0

Figura 6.29: Classifica dei migliori 3 agenti nel secondo trimestre 2011

Mese: Aprile	
Prodotto	Fatturato
Sugar EE Subscription	72500.0
Monitor LCD HD	56600.0
Monitor TFT XA	47300.0

Mese: Maggio	
Prodotto	Fatturato
Monitor LCD HD	64500.0
Sugar EE Subscription	56900.0
Server Power TT	44900.0

Mese: Giugno	
Prodotto	Fatturato
Server Power TT	50300.0
Sugar EE Subscription	43300.0
Monitor LCD HD	39900.0

Figura 6.30: Classifica dei migliori 3 prodotti nel secondo trimestre 2011

L'interrogazione di Figura 6.31, relativa ai tre agenti con il migliore fatturato, si compone di tre distinte query annidate, rintracciabili dall'operatore

SELECT. La query più interna ha il compito di restituire una tabella contenente il totale del fatturato di ogni agente in ogni mese secondo un ordinamento decrescente (ottenuto con la clausola DESC) in base al fatturato.

```

SQL Query Area
1 SELECT l.mese as MESE, l.agente as AGENTE, l.fatturato as FATTURATO, count(*) as CONTEGGIO
2 FROM
3   (SELECT a.id_agente, t.id_time, t.nomemese as mese, concat(a.cognome, ' ', a.nome)
4     as agente, sum(importo_co) as fatturato
5   FROM fact_ricavo f, agente a, time t
6   WHERE f.id_agente=a.id_agente and f.id_time=t.id_time and t.anno=2011
7   GROUP BY t.id_time, a.id_agente
8   ORDER BY t.id_time, fatturato DESC)
9
10  as l LEFT OUTER JOIN
11
12  (SELECT a.id_agente, t.id_time, t.nomemese as mese, concat(a.cognome, ' ', a.nome)
13    as agente, sum(importo_co) as fatturato
14  FROM fact_ricavo f, agente a, time t
15  WHERE f.id_agente=a.id_agente and f.id_time=t.id_time and t.anno=2011
16  GROUP BY t.id_time, a.id_agente
17  ORDER BY t.id_time, fatturato DESC)
18
19  as r ON l.id_time=r.id_time and l.fatturato>=r.fatturato
20
21 GROUP BY l.id_time, l.id_agente
22 HAVING CONTEGGIO>3
23 ORDER BY l.id_time, l.fatturato DESC

```

Figura 6.31: Query SQL per il grafico di Figura 6.29

La Figura 6.32 mostra una porzione di tale tabella relativamente al solo mese di *gennaio*.

id_agente	id_time	mese	agente	fatturato
7	4049	Gennaio	Olliver Chris	44100
5	4049	Gennaio	Jensen Max	37800
6	4049	Gennaio	Westin Will	32000
3	4049	Gennaio	Bronsen Sally	25500
2	4049	Gennaio	Smith Sarah	16400
1	4049	Gennaio	Brennan Jim	10500

Figura 6.32: Output della prima query annidata di Figura 6.31

Questa sarà successivamente messa in giunzione, tramite l'utilizzo del comando LEFT OUTER JOIN, con una tabella identica a se stessa (creata dalla seconda SELECT annidata), sfruttando il valore identificativo del mese (denominato *id_time*).

L'output derivante da questa giunzione restituisce per ciascun mese dell'anno, un record della prima tabella tante volte quante questo record (cioè l'agente) ha il fatturato maggiore o uguale a quello degli altri record (agenti) dello stesso mese (controllo *left.fatturato* >= *right.fatturato*).

Ne deriva che, se per un determinato mese, un generico agente mostra al suo attivo il peggior fatturato di vendita rispetto ai suoi colleghi, questo

comparirà una sola volta nella tabella dopo la giunzione; comparirà invece due volte se mostra il penultimo fatturato, tre volte se mostra il terzultimo e così via. La Figura 6.33 mostra un estrapolato del risultato ottenuto dopo il comando di giunzione tra le tabelle in questione.

id_agente	id_time	mese	agente	fatturato	id_agente	id_time	mese
7	4049	Gennaio	Olliver Chris	44100	7	4049	Gennaio
7	4049	Gennaio	Olliver Chris	44100	5	4049	Gennaio
7	4049	Gennaio	Olliver Chris	44100	6	4049	Gennaio
7	4049	Gennaio	Olliver Chris	44100	3	4049	Gennaio
7	4049	Gennaio	Olliver Chris	44100	2	4049	Gennaio
7	4049	Gennaio	Olliver Chris	44100	1	4049	Gennaio
5	4049	Gennaio	Jensen Max	37800	5	4049	Gennaio
5	4049	Gennaio	Jensen Max	37800	6	4049	Gennaio
5	4049	Gennaio	Jensen Max	37800	3	4049	Gennaio
5	4049	Gennaio	Jensen Max	37800	2	4049	Gennaio
5	4049	Gennaio	Jensen Max	37800	1	4049	Gennaio
6	4049	Gennaio	Westin Will	32000	6	4049	Gennaio
6	4049	Gennaio	Westin Will	32000	3	4049	Gennaio
6	4049	Gennaio	Westin Will	32000	2	4049	Gennaio
6	4049	Gennaio	Westin Will	32000	1	4049	Gennaio
3	4049	Gennaio	Bronsen Sally	25500	3	4049	Gennaio
3	4049	Gennaio	Bronsen Sally	25500	2	4049	Gennaio
3	4049	Gennaio	Bronsen Sally	25500	1	4049	Gennaio
2	4049	Gennaio	Smith Sarah	16400	2	4049	Gennaio
2	4049	Gennaio	Smith Sarah	16400	1	4049	Gennaio
1	4049	Gennaio	Brennan Jim	10500	1	4049	Gennaio

Figura 6.33: Output della seconda query annidata di Figura 6.31

In ultimo, la query più esterna ha lo scopo di raggruppare nuovamente i record sia sul valore identificativo dell'agente che del mese (rispettivamente *id_agente* e *id_mese*) al fine di contare il numero di occorrenze in cui questo compare (sfruttando la funzione `COUNT(*)`), determinando quindi la classifica desiderata per gli agenti.

MESE	AGENTE	FATTURATO	CONTEGGIO
Gennaio	Olliver Chris	44100	6
Gennaio	Jensen Max	37800	5
Gennaio	Westin Will	32000	4
Febbraio	Bronsen Sally	52000	6
Febbraio	Olliver Chris	52000	6
Febbraio	Jensen Max	48300	4
Marzo	Westin Will	50400	6
Marzo	Olliver Chris	50200	5
Marzo	Jensen Max	46500	4

Figura 6.34: Output finale della query di Figura 6.31

Per concludere vengono restituiti soltanto i record che presentano il valore `CONTEGGIO` maggiore o uguale a quattro (sfruttando la clausola `HAVING`

conteggio > 3), ottenendo come risultato finale i migliori tre agenti di vendita per ciascun mese sulla base dell'ammontare del fatturato. In Figura 6.34 è mostrato l'output conclusivo per la query di Figura 6.31 relativo ai soli mesi di gennaio, febbraio e marzo dell'anno 2011.

La funzione qui implementata non corrisponde alla funzione analitica `RANK()` ma vuole essere una metodologia per rispondere al report richiesto dal committente.

Alcune considerazioni che i responsabili possono trarre dalla consultazione delle informazioni riportate su questi ultimi due report, sono: riguardo al personale la possibilità di elargire premi agli agenti di vendita che sono rimasti in classifica per un certo numero di mesi; mentre per quel che concerne l'area commerciale l'eventualità di variare il portafoglio prodotti dell'azienda ad esempio eliminando i prodotti che durante l'anno non sono mai entrati in classifica o incentivandone il consumo e l'acquisto da parte degli acquirenti sfruttando opportune tecniche di marketing.

6.4 Rapporto per l'analisi del punto di pareggio

A conclusione di questo Capitolo si passa in rassegna la trattazione dei report inerenti l'analisi del punto di pareggio, che scaturiscono dai dati mantenuti nella tabella di Figura 4.6. Come discusso nei Capitoli precedenti, questo tipo di grafico utilizza un modello classico di rappresentazione a linee, rispettivamente per riportare sul piano cartesiano i valori dei dati per i costi fissi (retta di colore giallo), costi variabili (retta di colore arancio), costi totali (retta di colore rosso) e ricavi totali (retta di colore verde). Sull'asse delle ascisse sono mantenuti i giorni dell'anno di riferimento mentre sull'asse delle ordinate le quantità monetarie. Nell'esempio di Figura 6.35 si mostra il grafico del punto di pareggio riferito allo sola sede di *New York*.

Si nota come questo grafico sia fortemente basato sulle indicazioni grafiche presentate precedentemente in Figura 2.6. La data stimata per il punto di pareggio della sede di riferimento è riportata in alto; questo per favorire la ricezione immediata dell'informazione evitando che l'utente debba trar-

re questa conclusione manipolando il grafico (nell'esempio riportato la data stimata per il raggiungimento del punto di pareggio è prevista per il 15-08-2011). L'informazione principale coincide con l'intersezione della retta di colore rosso (che rappresenta i costi totali aziendali) con la retta di colore verde (che rappresenta i ricavi totali aziendali).

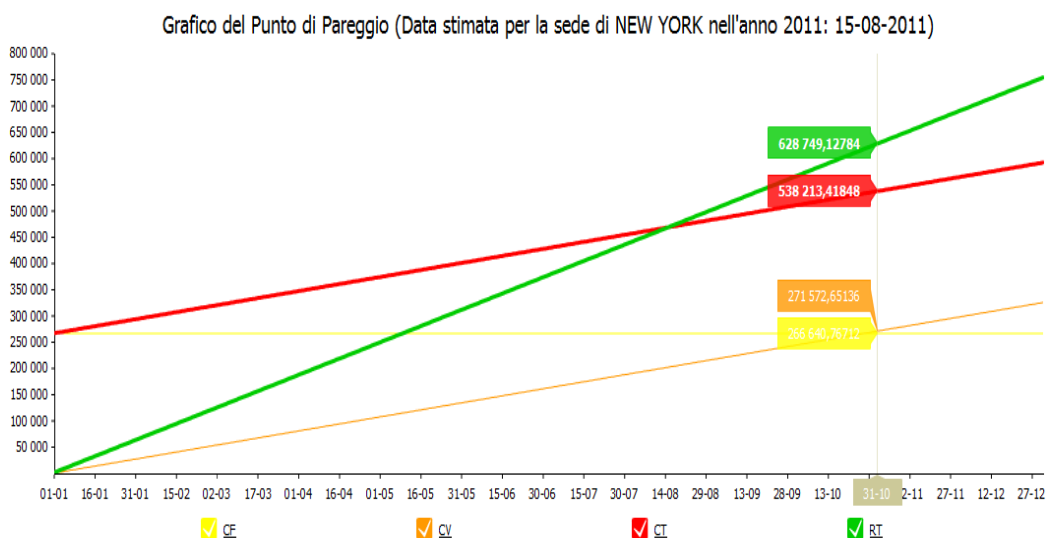


Figura 6.35: Esempio di grafico del punto di pareggio per sede

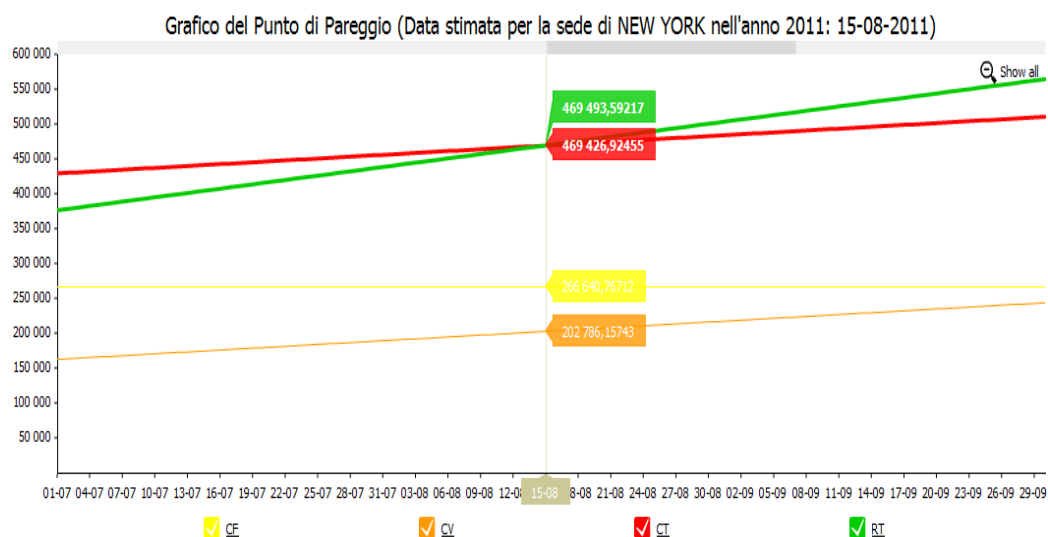


Figura 6.36: Dettaglio sul terzo trimestre

Il grafico è fortemente manipolabile dall'utente in quanto può decidere di

escludere parte delle informazioni dalla rappresentazione, ad esempio eliminando le rette rappresentanti l'ammontare dei costi fissi e l'ammontare dei costi variabili (come mostrato in Figura 6.37), oppure ottenendo una rappresentazione più dettagliata delle informazioni usufruendo dell'apposito strumento di zoom incorporato nel grafico.

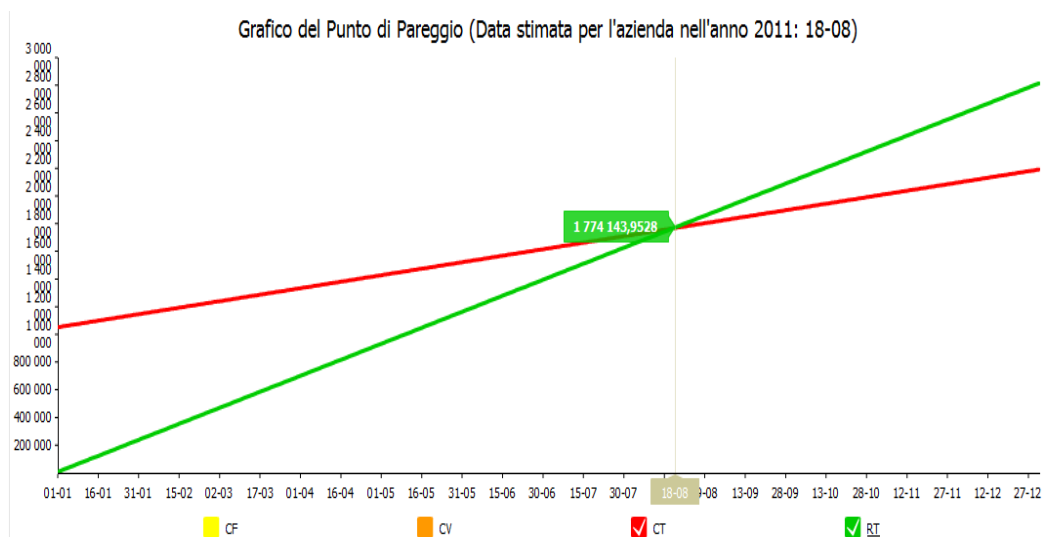


Figura 6.37: Esempio complessivo su tutte le sedi

Quest'ultimo è il caso dell'esempio di Figura 6.36 in cui si mostra il dettaglio dell'andamento delle rette dei costi e di ricavo per il terzo trimestre dell'anno. I dati di riferimento sugli assi del piano cartesiano vengono dinamicamente e opportunamente modificati a seconda del dettaglio che si vuole rappresentare (in questo esempio le date riportate sull'asse delle ascisse vanno dal 1-7-2011 al 30-9-2011 perchè riferito al terzo trimestre). Qualora l'utente volesse estrarre il dato puntuale del valore complessivo del costo e del ricavo totale in una certa data dell'anno, semplicemente posizionando il cursore sulla data prescelta (ad esempio il 15-8-2011 come in Figura 6.36), i valori monetari appaiono all'interno di appositi box in corrispondenza di ciascuna retta del grafico.

Lo stesso report può essere creato, a discrezione dell'utente, per ciascuna sede (come mostrato in precedenza) o alternativamente può essere definito sfruttando la complessità dei dati di tutte le sedi aziendali (come mostrato in Figura 6.37).

6.5 Osservazioni conclusive

Si riportano alcune considerazioni in relazione all'utilizzo di MySQL, nella versione specificatamente indicata, per lo sviluppo del progetto.

Nome	Descrizione
WITH ROLLUP	Comporta la creazione di più raggruppamenti considerando le combinazioni ottenute dalla lista di attributi del GROUP BY eliminando un attributo alla volta dalla destra alla sinistra; per la lista vuota tutti i dati fanno parte di un unico raggruppamento. Ad esempio, GROUP BY (anno, mese) WITH ROLLUP corrisponde all'esecuzione di tre raggruppamenti: GROUP BY (anno, mese), GROUP BY (anno), GROUP BY ().
AVG()	Restituisce il valore medio dell'argomento
COUNT()	Restituisce un conteggio di un certo numero di righe ottenute
COUNT(DISTINCT)	Restituisce il conteggio di un certo numero di valori diversi
GROUP CONCAT()	Restituisce una stringa concatenata
MAX()	Restituisce il valore massimo dell'argomento
MIN()	Restituisce il valore minimo dell'argomento
STD()	Restituisce la deviazione standard dell'argomento
SUM()	Restituisce la somma di valori
VARIANCE()	Restituisce la varianza standard dell'argomento

Tabella 6.1: Funzioni e modificatori nella GROUP BY in MySQL v.5.0

In Tabella 6.1 sono mostrate parte delle clausole e funzioni che la base di dati in oggetto supporta all'interno dell'operatore GROUP BY.⁴ Si nota come alcuni degli operatori OLAP, tra i quali il classico CUBE(), supportati su alcune delle basi di dati tipo ORACLE, DB2 e SQLServer non siano previsti nell'insieme di MySQL.

Pertanto, utilizzando quest'ultima base di dati, qualora si rende necessario l'utilizzo di tale operatore in una query SQL, lo sviluppatore è tenuto a fornire il medesimo risultato sfruttando opportunamente la combinazione

⁴Per la visualizzazione dell'elenco completo di tutti gli operatori e le funzioni supportate da MySQL versione 5.0 si rimanda alla pagina web: <http://dev.mysql.com/doc/refman/5.0/en/functions.html>

di molteplici query SQL caratterizzate da un certo insieme di operatori e funzioni supportate.

Nome	Descrizione
ABS()	Restituisce il valore assoluto
ADDTIME()	Aggiunge la data nell'argomento
AVG()	Restituisce il valore medio dell'argomento
BETWEEN..AND	Controlla la presenza di un valore in un certo intervallo di valori
CASE	Operatore per la distinzione dei casi
CAST()	Esegue il CAST di un valore come un certo tipo
CONCAT()	Restituisce una stringa concatenata
COUNT()	Restituisce il conteggio del numero di righe restituite
DATE()	Estrae la data da un'espressione DATETIME
JOIN	Seleziona dati prelevandoli fra più tabelle, evidentemente correlate tra loro. Esistono le INNER JOIN, OUTER JOIN e CROSS JOIN. In particolare le OUTER JOIN oltre ad estrarre i dati sulla base di corrispondenze tra tabelle, estraggono anche le righe di una tabella che non hanno corrispondenti nell'altra. Nella sintassi di MySQL, la parola OUTER è facoltativa: scrivere LEFT JOIN o LEFT OUTER JOIN è equivalente. Lo stesso vale per RIGHT JOIN o RIGHT OUTER JOIN. In MySQL non è possibile effettuare le FULL OUTER JOIN.
LIKE	Semplice schema di corrispondenza nell'argomento
MAX(), MIN()	Restituiscono il valore massimo e minimo
ROUND()	Arrotonda l'argomento
SUBSTR()	Restituisce la sottostringa come specificato
UNION()	Combina i risultati provenienti da più SELECT in un unico set di risultati

Tabella 6.2: Funzioni analitiche e operatori in MySQL v.5.0

La Tabella 6.2 riporta invece una porzione delle funzioni analitiche che si possono trovare in MySQL e che maggiormente sono state utilizzate nelle query mostrate nelle Sezioni precedenti.

La particolarità dello strumento che maggiormente risalta nell'elenco ap-

pena presentato è che non vi è alcun riferimento alle più classiche funzioni analitiche con finestra. Nello specifico non sono supportate le funzioni CUME_DIST(), RANK(), DENSE_RANK(), PERCENT_RANK() e ROW_NUMBER() comprese invece nel pacchetto fornito da ORACLE e in parte su DB2.

6.6 Riassunto

- Si mostra un tipico esempio di sviluppo di un documento di report tramite l'utilizzo dello strumento Report Designer e alternativamente tramite l'utilizzo di opportune tecniche di programmazione che sfruttano le sequenze d'azione.
- Data una sintesi sulle varianti possibili di presentazione dei risultati sottoforma di rapporti si analizzano nel dettaglio alcuni esempi che sono stati realizzati per questo progetto, utilizzando la classificazione che distingue i rapporti generati a partire dall'esecuzione di query SQL relativamente semplici, da query SQL relativamente complesse e in ultimo quelli generati da query SQL complesse.
- Per ciascuno degli esempi mostrati si ricorre ad una breve descrizione di tutti i passaggi di sviluppo compresi tra la definizione dell'interrogazione e la presentazione finale dell'output corrispondente sottoforma di report grafico o tradizionale.
- Si conclude l'argomento con alcune considerazioni aggiuntive in relazione alle capacità degli strumenti utilizzati di far fronte alle richieste progettuali, di analisi e di sviluppo desiderate.

Capitolo 7

Le tecnologie Open Source utilizzate

In questo Capitolo conclusivo si presenta una visione d'insieme di tutti gli strumenti e le tecnologie utilizzate per lo sviluppo del progetto. Si descrivono gli strumenti principali che compongono la suite Pentaho Community Edition, la piattaforma di business intelligence utilizzata, e tutte le componenti che a vario titolo sono state interessate.

7.1 Risorse e relazioni del sistema

Open source (termine inglese che significa sorgente aperto) indica un software i cui autori (più precisamente i detentori dei diritti) ne permettono, anzi ne favoriscono il libero studio e l'apporto di modifiche da parte di altri programmatori indipendenti. Questo è realizzato mediante l'applicazione di apposite licenze d'uso.¹

L'intero progetto è stato sviluppato sfruttando le potenzialità di alcuni software open source. Nelle Sezioni successive si presentano le principali risorse utilizzate, descrivendone brevemente le caratteristiche fondamentali e gli strumenti che queste offrono agli sviluppatori al fine di raggiungere i propri obiettivi.

¹Definizione tratta da <http://it.wikipedia.org>

In Figura 7.1 si mostra come queste risorse influiscono sul risultato finale e come si relazionano. Lo strumento di Business Intelligence, utilizzato dagli sviluppatori, si colloca ad un livello più alto e sfrutta le potenzialità dei sistemi che stanno ai livelli inferiori. Riceve i dati di input tramite connessioni ODBC sia dal *CRM operativa* (SugarCRM) sia dal database *BudgetOLTP* (di cui ne discuteremo i dettagli in seguito). Le informazioni, opportunamente trattate e trasformate costituiranno la fonte informativa delle analisi che verranno messe a disposizione degli utenti. Gli utilizzatori finali del sistema si collocano al livello più basso della gerarchia ed avranno accesso ai loro dati, alle analisi, ai report e alle dashboard online tramite l'applicativo *PortaBI*. L'accesso sarà garantito da un nuovo plugin che verrà opportunamente reso disponibile all'utenza direttamente sull'applicativo.



Figura 7.1: Livelli del sistema

7.1.1 La Suite Pentaho Community Edition

Si considera Pentaho come una suite per la Business Intelligence piuttosto che un singolo prodotto, infatti è costituito da una collezione di programmi che lavorano insieme per creare e distribuire soluzioni per un certo Business. Alcuni di questi componenti forniscono delle funzionalità considerate di base, come ad esempio le autenticazioni utente o la gestione delle connessioni ai databases. Altri componenti, al contrario, offrono diverse funzionalità ad un livello più alto, come ad esempio la visualizzazione dei dati tramite tabelle e grafici.

Spesso le componenti che offrono funzionalità ad alto livello contano su altre componenti che invece danno un supporto ad un livello più basso. L'insieme dei programmi che compongono l'intera suite può essere considerato pertanto come una pila di componenti in cui ciascun livello è pensato per far avvicinare la funzionalità all'utente finale. In questa visione è quindi più ragionevole parlare di *Pentaho BI stack*, mostrata in Figura 7.2.

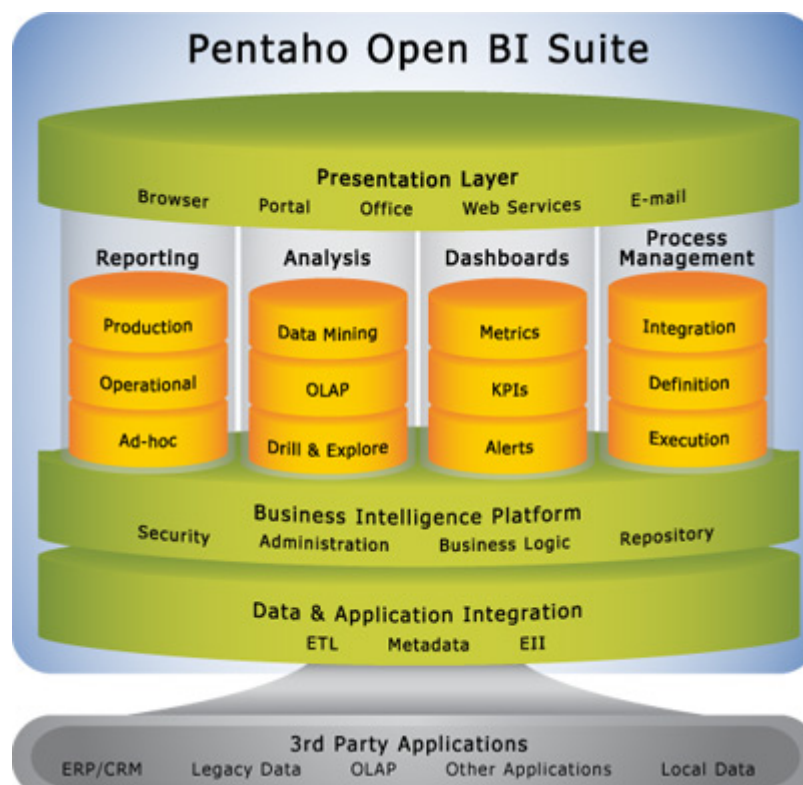


Figura 7.2: Struttura a livelli di Pentaho BI

In figura sono rappresentati i principali livelli della pila, il vertice è occupato dal *presentation layer* mentre sul fondo si colloca il *data and application integration layer*. L'utente interagisce con il *presentation layer* che può assumere svariate forme, delle quali la più comune è senz'altro l'accesso tramite un web browser. Le principali aree funzionali denominate reporting, analysis, dashboards e process management costituiscono il corpo centrale della pila, considerando il fatto che la piattaforma stessa offre funzionalità di base per la sicurezza e l'amministrazione. Il data integration layer completa la pila ed è necessario per ottenere dati provenienti da diversi sistemi sorgente in un ambiente condiviso data warehouse. Naturalmente questo tipo di architettura definisce a grandi linee la struttura di Pentaho e il profilo di una possibile soluzione, ma non necessariamente definisce come la struttura deve essere costruita. Pertanto è lasciata allo sviluppatore la libertà di poter mettere insieme molteplici componenti e di poter utilizzare solamente quelli che più si avvicinano alle esigenze e agli obiettivi che si vogliono raggiungere.

I programmi che compongono *Pentaho BI suite* possono essere classificati secondo criteri e punti di vista differenti di cui ne riportiamo i principali [Bouman 09]:

- **Funzionalità**

Un primo modo per categorizzare i vari componenti è tramite la loro funzionalità. Con questo termine si intende l'attività o le attività per le quali un programma è stato progettato. Dalla prospettiva dell'utente la funzionalità è ciò che definisce lo scopo del programma stesso e a tale proposito Pentaho offre, tra le molteplici funzionalità, quelle classiche di Business Intelligence quali *ETL*, *Reporting* e *OLAP*. Le componenti che forniscono la suite di questi strumenti nel loro complesso sono conosciuti come *Pentaho platform*.

- **Server, Web Client e Desktop**

Un'altro criterio di distinzione classifica i programmi come Client, Server o Desktop. Vengono etichettati come Server quelli che tipicamente sono eseguiti su una macchina centrale accessibile dai *web client* tramite reti intranet o internet. Tutte le altre applicazioni che non possono

essere considerate programmi Server vengono definite come desktop, tipicamente risiedenti sulle macchine locali degli utenti. Sono queste le applicazioni che progettisti e sviluppatori utilizzano per lanciare richieste ai programmi Server con il quale sono collegati. Un tipico esempio di distinzione tra applicazioni Server, client e desktop è la progettazione, la pubblicazione e l'esecuzione di un documento di report. In questo caso *Report Designer* è l'applicazione desktop utilizzata per creare il report che sarà pubblicato sul *Pentaho Server*. A sua volta quest'ultimo potrà eseguire il report su richiesta utilizzando l'applicazione *Report Engine*, mentre il risultato verrà visualizzato come output grazie a *Pentaho Web Portal* che implementa il client.

- **Front end e back end**

Tipicamente si classificano le applicazioni front end come l'insieme dei programmi che forniscono un'interfaccia di tipo *user friendly*, capaci quindi di facilitarne l'uso da parte degli utenti. Al contrario, il termine back end è interpretato come l'insieme dei programmi che non sono riservati all'interazione con l'utente, ma piuttosto destinati a ricevere i comandi da qualche strumento di front end che sa tradurre l'azione dell'utente in operazioni su cui il back end è in grado di operare ed esercitare le funzioni reali corrispondenti. Anche se può sembrare che la distinzione tra back end e front end sia fortemente legata alla classificazione precedentemente descritta, in quanto le applicazioni client e desktop coincidono spesso con quelle front end, è altrettanto possibile implementare un'applicazione back end come una di tipo desktop. Naturalmente all'interno della suite Pentaho la maggior parte delle sue componenti non prevedono un'interfaccia accessibile dagli utenti, conseguentemente non hanno una parte front end.

- **Tecnologia di base**

Quasi tutti i programmi contenuti in Pentaho sono stati definiti utilizzando il linguaggio di programmazione *Java*, mentre alcune componenti lato server implementano applicazioni web basate su tecnologia *AJAX*. Questa distinzione, a differenza delle precedenti, può risultare irrilevan-

te per l'utente finale, ma comunque utile per amministratori di sistema e sviluppatori. Questa puntualizzazione assume comunque un aspetto importante che risiede nel fatto che una delle proprietà più rilevanti della piattaforma Java riguarda l'estrema flessibilità e portabilità delle sue applicazioni sulle più svariate architetture hardware e sistemi operativi. Conseguentemente anche Pentaho è utilizzabile su differenti sistemi operativi.

Ritornando sulla Figura 7.2 è importante a questo punto commentare nel dettaglio il presentation layer. Pentaho è dotato di una interfaccia web chiamata *user console* che costituisce la parte front end grazie alla quale l'utente può interagire con il server. Tipicamente il presentation layer viene utilizzato per la lettura e la visualizzazione dei contenuti esistenti quali reports, dashboards e analysis ma lo si può anche utilizzare per creare nuovi contenuti di Business Intelligence. Pertanto sulla user console lo sviluppatore avrà a disposizione tutte le applicazioni che in precedenza sono state definite come Desktop. In Tabella 7.1 sono riportate le applicazioni desktop utilizzate dall'utente ed i rispettivi componenti che implementano tali funzionalità lato server.

DESKTOP TOOL	SERVER/BI COMPONENTS
Design Studio (PDS)	BI Platform
Metadata Editor (PME)	Metadata layer, Ad hoc Reporting component
Schema Workbench (PSW)	OLAP Engine
Aggregate Designer (PAD)	OLAP Engine
Report Designer (PRD)	Reporting engine
Data Integration (PDI)	ETL engine
Weka	Data Mining engine

Tabella 7.1: Corrispondenze tra Desktop tool e componenti lato server

Nel seguito si descrivono le *Desktop tool* [CWH 11] prettamente utilizzate per lo sviluppo del progetto e i cui risultati sono presentati nei Capitoli successivi:

- **Design Studio**

Tipicamente lo strumento Design Studio non è utilizzato per creare

nuovi contenuti ma piuttosto per creare *workflows* che sfruttano i concetti di Business Intelligence esistenti.

- **Metadata Editor**

Metadata Editor permette di costruire strati di metadati in grado di fornire un livello di astrazione tra un database relazionale e l'utente finale. Questa applicazione permette di raccogliere gli oggetti utili per l'utente, come ad esempio il nome di un cliente, la nazione, una quantità monetaria ecc. e tradurre la selezione in un linguaggio SQL necessario per il recupero delle informazioni da un database.

- **Schema Workbench**

Schema Workbench lo si può considerare come il costruttore degli schemi multidimensionali (cubi OLAP). In realtà questa applicazione ha come funzione principale quella di tradurre ciò che l'utente progetta in linguaggio *MDX* (Multidimensional EXpression). Solo nel passo successivo il *file.mdx* verrà interpretato dal motore *Mondrian Engine*, un motore *ROLAP* basato su linguaggio java, in grado di creare una vista multidimensionale dei dati risiedenti su un database.

- **Aggregate Designer**

Aggregate Designer è lo strumento utilizzato per la progettazione automatica di tabelle aggregate che saranno utilizzate dal motore *Mondrian* al fine di migliorare le prestazioni delle operazioni sui cubi OLAP.

- **Report Designer**

Report Designer è l'applicazione utilizzata per la costruzione di report per la piattaforma Pentaho. Presupposto che la comunicazione viene considerata un nucleo necessario di Business Intelligence e di fondamentale importanza per le aziende, Report Designer consente alle organizzazioni facile accesso e distribuzione delle informazioni sia verso i dipendenti che verso clienti e partner. Lo strumento consente anche di generare documenti di reportistica su molteplici formati di output quali ad esempio PDF, HTML, RTF e tradizionali fogli di calcolo in modo

da facilitarne la distribuzione via web o tramite e-mail.

- **Data Integration**

Data Integration è lo strumento che permette agli sviluppatori di definire flussi ETL intesi come sequenze di trasformazioni su dati in ingresso provenienti da fonti eterogenee. Un insieme di trasformazioni possono essere raccolte in un *job* e a sua volta in uno stesso progetto possono coesistere più *job* in cui l'input di uno è fornito dall'output dell'altro. Questo tipo di funzionalità permette agli sviluppatori di poter separare diverse parti di un unico e complesso flusso ETL in modo da favorirne le eventuali modifiche e i successivi cambiamenti che in fase di progettazione possono avvenire. Tipicamente lo strumento di Data Integration nella suite Pentaho è denominato *Spoon*.

- **Weka**

Weka è lo strumento previsto nella suite Pentaho come soluzione alla necessità di aggiungere un'applicazione apposita per il data mining. Il Data mining è la funzionalità che elabora i dati attraverso complessi algoritmi per scoprire modelli significativi e le correlazioni che altrimenti potrebbero essere nascoste con l'analisi e il reporting standard. Le informazioni ottenute possono essere usate per aiutare a comprendere meglio il business aziendale e per migliorare le prestazioni future attraverso l'analisi predittiva.

7.1.2 SugarCRM, il Customer Relationship Management

Un'altra risorsa open source ampiamente utilizzata per lo sviluppo del progetto è il Customer Relationship Management *SugarCRM*. Esso è infatti il CRM operativo integrato sul prodotto gestionale PortaBI (di cui discuteremo nella Sezione successiva) sul quale gli amministratori aziendali organizzano, popolano e conservano in modo efficiente tutte le informazioni riguardanti le relazioni con i propri clienti. È pensato per gestire tutto ciò che gravita attorno al Business aziendale, tra cui: trattative commerciali, offerte,

anagrafiche clienti e contatti, campagne di marketing, assistenza, calendario condiviso, gestione email [SGR 11].

La particolarità di questo CRM risiede nella sua possibilità di essere organizzato per moduli, ciascuno dei quali rappresenta uno specifico aspetto funzionale importante ai fini della gestione aziendale. Infatti ciascun modulo, considerato separatamente, mantiene i dettagli principali riferiti ad un certo cliente, ma se si considera la totalità dei moduli e le loro relazioni si può costruire un intero ciclo di vita di ciascuna entità. Tale ciclo di vita ha inizio con il primo inserimento di dati e delle informazioni utili all'interno del sistema, che nella maggior parte dei casi saranno informazioni parziali rispetto alla totalità richiesta, e si conclude con il raggiungimento dello stato di fidelizzazione del cliente e conseguentemente di tutte le opportunità che si possono generare da tale traguardo.

Nel suo complesso, SugarCRM è composto dai seguenti moduli: Home, Leads, Contacts, Accounts, Opportunities, Activities, Emails, Reports, Cases, Bug Tracker, Knowledge Base, Documents, Products, Forecasts, Contracts, Projects, Employees. Di questi moduli solo alcuni sono stati impiegati per lo sviluppo del progetto:

- **Lead**

Un Lead rappresenta un contatto manifestatosi nelle prime fasi di un processo di vendita. Per questa ragione tipicamente un Lead è caratterizzato da una parzialità di informazioni dovute al loro mancato reperimento. Esso può essere associato ad una qualsiasi *campaigns* al solo scopo di poter monitorare la capacità di quest'ultime di generare nuovi Leads.²

²Ad oggi la letteratura sull'argomento non esplica esaustivamente la differenza tra un'entità *lead* e un'entità *prospect*. Si riporta pertanto una definizione tratta da <http://www.picrmsoftware.com.au> secondo cui: "Un lead è in genere un individuo che ancora non ha dato il permesso di poter entrare in contatto con lui a tutti gli effetti". Nelle classiche fasi di vendita generalmente lo potremmo collocare nella fascia di clienti di cui si conosce poco o niente di loro ed anche se alcuni potrebbero essere provvisoriamente interessanti a ciò che l'azienda ha da offrire, purtroppo questi non hanno ancora preso contatto con i manager o l'azienda stessa. Un lead potrebbe includere, tra le sue svariate forme, un biglietto da visita, un elenco di contatti acquistato, dati tratti da fonti quali le Pagine Gialle o Google. Per concludere, un lead può conoscere l'azienda ma non ha ancora

- **Contact**

Un contact rappresenta una qualsiasi persona fisica, tipicamente un responsabile di vendita di un'azienda cliente. Viene considerato come un vantaggio acquisito di cui se ne può misurare e valutare una possibile occasione di vendita. Esso può essere associato ad una *opportunity* o ad una *campaign*, il che permette di poter monitorare la capacità di quest'ultima di poter generare nuove opportunità per l'organizzazione. Rappresenta lo stadio successivo di crescita di un lead conseguente ad un nuovo incontro e al reperimento delle informazioni mancanti.

- **Account**

Un Account mantiene tutte le informazioni disponibili sui clienti, quali ad esempio nome, indirizzo e recapito telefonico. In questo modulo col termine cliente si intende l'azienda cliente, e non la persona fisica con la quale è stato creato il contatto che invece risiede sul modulo contact. Naturalmente queste due informazioni saranno associate nella base di dati permettendo quindi di poter risalire da un account ad un contact ma anche ad una opportunity.

- **Opportunity**

Una opportunity specifica un importo stimato per un determinato accordo di vendita con un contact riconducibile ad un account. Si possono considerare come delle entità autonome all'interno della base di dati per le quali l'amministratore le definisce e regola a seconda delle esigenze di vendita. Tipicamente l'importo monetario assegnato può variare sulla base della fase di vendita e delle trattative in corso.

- **Email**

Il modulo smail implementa un client di posta elettronica in grado di gestire le mail in entrata e in uscita. La sua importanza risiede soprattutto nella possibilità di tener traccia di tutti gli utenti contattati per

concesso i diritti per fare in modo che quest'ultima si metta in contatto con lui e non ha dato il permesso di poter essere aggiunto alla *mailing list* aziendale.

il lancio e la presentazione di un nuovo evento o di una nuova *Campaign*.

- **Campaign**

Una *campaign* rappresenta un evento di marketing che si rivolge ad un vasto gruppo di individui e organizzazioni clienti. Il processo di definizione di una *Campaign* ha origine con l'individuazione degli obiettivi che si vogliono raggiungere tramite l'evento e termina con l'identificazione di un insieme target a cui questo si rivolge. La *Campaign* è da considerare l'elemento centrale tra tutte le entità menzionate al quale è possibile associare molteplici lead, contact, account e conseguentemente le opportunità. Sono previste svariate forme di *campaign*, tra cui le *E-mail*, le *newsletter* e *web* che sono quelle sulle quali si è maggiormente concentrata l'attenzione per lo sviluppo del progetto. È da considerare una *campaign* di tipo e-mail se questa è basata sull'invio di messaggi di posta elettronica ad un certo target di clienti, tipicamente contenenti le informazioni principali sull'evento organizzato, gli obiettivi di interesse comune e sul prodotto presentato.

7.1.3 Lo strumento PortaBI

PortaBI è uno strumento di intelligence dedicato all'attività di vendita che mette insieme un Customer Relationship Management e la Business Intelligence oltre a strumenti di pianificazione e previsione. Al suo interno si trovano un'insieme di tecnologie open source che consentono principalmente di raccogliere e mantenere informazioni sui clienti, numeri di vendite, la redditività e altre informazioni utili. L'obiettivo finale è quello di far acquisire all'utente la capacità di prendere decisioni basate su informazioni in tempo reale direttamente dalla propria workstation.³

Nel suo complesso il prodotto si propone di facilitare la gestione e l'analisi di alcuni settori critici in ambito aziendale: la pianificazione delle vendite, l'analisi delle vendite, il monitoraggio delle campagne di vendite e l'analisi dei clienti.

³Tratto da <http://www.portabi.com/>

Per il raggiungimento di questo obiettivo PortaBI prevede come primo passo quello di memorizzare e organizzare i dati in modo tale che l'utente possa avere una visione completa di tutti i clienti, partner e fornitori sia potenziali che già acquisiti. Questa esigenza deriva dal fatto che spesso queste informazioni vengono mantenute su diversi supporti o molteplici basi di dati e quindi risulta più complesso eseguire operazioni di previsione di vendita, monitoraggio della redditività e soddisfazione del cliente, gestione delle spese di marketing ecc.

Il secondo aspetto interessante e gestito da PortaBI riguarda la possibilità di eseguire le analisi sui dati memorizzati. A tale proposito le informazioni necessarie dovranno essere raccolte all'interno di strutture dedicate, quali data warehouse o data mart, il cui popolamento avrà origine da flussi ETL in grado di estrarre, trasformare e presentare i dati nella forma che meglio si adatta al tipo di analisi che dovrà essere effettuata. Lo strumento prevede poi la possibilità di visualizzare i risultati di analisi attraverso il web presentandoli sottoforma di report o dashboard, comprensivi di tavole, grafici, immagini e fogli di calcolo.

Il software gestionale, oggetto di questa Sezione, diventerà il supporto front end sul quale l'utente finale potrà accedere al plugin appositamente dedicato per la gestione e pianificazione del processo di budget aziendale. Nei Capitoli successivi si affronta questa tematica ed i vari passaggi progetturali definiti per il raggiungimento dell'obiettivo.

7.2 Riassunto

- È mostrata un'ampia panoramica sullo strumento di BI, la Suite Pentaho C.E. La versione utilizzata è interamente open source e scaricabile in rete. Questa versione differisce per alcuni strumenti aggiuntivi che invece sono forniti nella versione ceduta tramite licenza a pagamento. In dettaglio si elencano tutti gli strumenti della Suite, le loro caratteristiche e la modalità di funzionamento.
- È presentata una breve introduzione al Customer Relationship Management in dotazione all'azienda cliente, anch'esso open source, descrivendone in modo puntuale gli elementi principali che lo compongono.

-
- Si descrive lo strumento PortaBI da considerare come l'anello di congiunzione tra il sistema di analisi progettato e l'utente finale. Definisce, in altri termini, l'interfaccia utente grazie alla quale quest'ultimo può interagire con il sistema, eseguire le analisi dei dati e consultare i risultati ottenuti.

Conclusioni

Questa tesi presenta il lavoro di sviluppo e progettazione di un data warehouse appositamente definito a supporto dei processi di analisi che hanno come obiettivo la redazione di un budget commerciale annuale per l'impresa. Si evidenzia l'importanza di particolari sistemi informativi aziendali utilizzati nel trattamento delle informazioni e come questi offrono un adeguato supporto a livello decisionale.

In particolare, è mostrato l'intero processo di sviluppo distinto per fasi di progettazione, ciascuna di queste opportunamente discussa nei singoli Capitoli che compongono la tesi. Si presenta inizialmente l'argomento principale di discussione, il budget, nelle sue varie forme e nelle sue implicazioni aziendali. Successivamente si affronta, tramite l'utilizzo di un classico esempio trattato in letteratura, la problematica derivante dall'unione delle esigenze aziendali inerenti alla stesura di un corretto piano di budget con quelle inerenti ad una corretta progettazione di una base di dati in grado di mantenere e restituire le informazioni utili. La fase che segue è incentrata sulla progettazione del data warehouse e tratta l'argomento scandendo le molteplici fasi di cui è composta (dalla corretta definizione del problema e raccolta dei requisiti di analisi alla progettazione logica). A seguire si affrontano le problematiche dovute al recupero, al trattamento e al popolamento dei dati che costituiranno la risorsa fondamentale per la quale il data warehouse è stato costruito. In conclusione si mostrano le analisi che è possibile eseguire sui dati di interesse attraverso una loro rappresentazione multidimensionale e come tali informazioni possono essere rappresentati in termini di documenti di reportistica e grafici prestazionali.

Anche se il progetto segue un andamento lineare tra le varie fasi affrontate, da come si osserva nella tesi, in realtà è opportuno precisare che i mesi

dedicati allo sviluppo sono stati soggetti a continui meccanismi di *feedback* e *feedforward* tra le varie fasi, ogni volta che si presentavano problematiche ad un determinato passo e la cui soluzione implicava una rivisitazione e modifica nel passo precedente. Un esempio classico può essere quello della progettazione dei singoli data mart. Alcune soluzioni elencate nel Capitolo che tratta l'argomento si riferiscono a soluzioni realmente implementate durante lo sviluppo ma che per ragioni di recupero informazioni oppure di creazione del modello multidimensionale per le analisi o di generazione della reportistica richiesta non si confanno alle esigenze volute.

Una volta ultimata l'interfaccia grafica utente (la cui progettazione non è argomento di questa tesi), il data warehouse può essere popolato delle informazioni riferite all'azienda committente che ha commissionato il lavoro; quindi tutte le analisi, i report e i grafici presentati possono trovare un riscontro reale di utilizzo ad uso di una specifica realtà d'impresa.

Il progetto si presta anche a ulteriori sviluppi e ampliamenti futuri. In primo luogo è possibile pensare ai report generati che sfruttano la classica metodologia di rappresentazione utilizzata frequentemente nei bilanci economici, ovvero presentando le medesime informazioni sia per l'anno corrente che per l'anno precedente. Si può quindi pensare ad una nuova categoria di report che mostrano il dettaglio dei costi aziendali per ciascuna sede nell'anno corrente confrontati con gli stessi riferiti all'anno precedente. Negli esempi mostrati nella tesi questo tipo di trattamento dati non è stato presentato per mancanza delle informazioni necessarie, dal momento che il nuovo sistema raccoglie i dati di un unico anno. Nei periodi a seguire, quando le informazioni mantenute nelle base di dati copriranno un arco temporale almeno biennale, sarà possibile definire i confronti citati.

Alternativamente è possibile ampliare il sistema concentrandosi sulla possibilità di sviluppare eventuali revisioni del budget che per questo progetto non sono state trattate, dando la possibilità all'utente di poter modificare e rivalutare i dati utili per definire scenari alternativi a fronte di differenti bisogni strategici o di influenze esterne positive o negative; il tutto prima di ottenere una versione ufficiale del budget.

Una volta maturata la giusta esperienza sul trattamento informazioni relative alla stesura del budget e raccolte e valutate le informazioni sui pregi

e difetti sul sistema realizzato, su eventuali inefficienze, mancanze o nuove esigenze scaturite dal suo utilizzo, è possibile avviare un nuovo progetto, che a partire da questo, si prefigge l'obiettivo di costruire un sistema di budget generico per un molteplice numero di imprese che operano su diversi settori, eventualmente lasciando un certo margine di personalizzazione. In questa ottica futura, il progetto presentato nella tesi, più che un lavoro fine a se stesso vuole essere un caso di studio; un punto di partenza e di valutazione delle varie problematiche affrontate per un'eventuale espansione del sistema ad altre imprese.

L'esperienza accumulata nel periodo interessato allo sviluppo del progetto ci ha permesso di utilizzare alcune delle tecnologie per la business intelligence applicate ad un caso reale. Abbiamo preso confidenza con il mondo lavorativo, nel caso specifico, nella modalità di lavoro in gruppo dove si incontrano analisti, programmatori, tecnici e progettisti specializzati in diversi settori al fine di conseguire un obiettivo comune. Soprattutto ci è stata data la possibilità di seguire l'intero sviluppo di un progetto che va dalle primissime fasi del suo concepimento a quelle della progettazione vera e propria, proponendone una possibile soluzione.

Appendice A

Ulteriori esempi di report e grafici

Si riportano ora ulteriori esempi, diversi dai precedenti mostrati nel Capitolo 6, di rapporti che è possibile ottenere utilizzando lo strumento Saiku e grafici prodotti con l'ausilio della tecnica dell'Xaction. Ciascun esempio è accompagnato dalla relativa interrogazione SQL utilizzata per il recupero dei dati dal data warehouse e da alcune considerazioni in merito. Tutti gli esempi mostrati sono il risultato di specifiche richieste del committente.

Rapporti basati sui requisiti di analisi

Sulla base dei requisiti di analisi numero 1 e 7 si generano i report mostrati nelle Figure A.1 e A.2 strutturati secondo le esigenze espresse dal committente.

1	Importo totale della voce di costo a budget per sede e per agente, per mese (trimestre, semestre e per l'intero anno corrente)
7	Importo totale della voce di costo consuntivata per sede, per agente, per mese (trimestre, semestre e per l'intero anno corrente)

Nel primo caso sono mostrati gli importi totali di costo sia a budget che a consuntivo (e il derivante ammontare dello scostamento) delle singole voci di costo per ciascuna sede, per trimestri dell'anno in corso. Nell'esempio specifico inoltre, sono stati applicati opportuni filtri (sulla clausola WHERE nella

query SQL) che restituiscono i valori delle misure di interesse per le sole voci di costo *energia elettrica* e *pulizia uffici* e per i soli trimestri *T1* e *T2*.

		2011					
		T1			T2		
nome	nome	budget_c	consuntivo_c	scostamento_c	budget_c	consuntivo_c	scostamento_c
Energia elettrica	MADRID	5.800	6.390	590	6.000	6.530	530
	MILANO	5.200	5.640	440	5.400	5.930	530
	MONACO	6.100	6.270	170	6.300	6.440	140
	NEW YORK	6.200	6.410	210	6.600	6.360	-240
	TORONTO	6.400	6.340	-60	6.600	6.690	90
Pulizia ufficio	MADRID	4.200	4.170	-30	4.500	4.450	-50
	MILANO	2.800	2.950	150	3.000	3.090	90
	MONACO	3.400	3.420	20	3.600	3.700	100
	NEW YORK	3.400	3.150	-250	3.600	3.490	-110
	TORONTO	3.600	3.500	-100	3.750	3.740	-10

Figura A.1: Report per i requisiti 1 e 7 per sede

Nel secondo caso sono mostrati gli importi totali di costo sia a budget che a consuntivo (e il derivante ammontare dello scostamento) delle singole voci di costo per ciascun agente di vendita, per semestre dell'anno in corso. Anche in questo specifico esempio sono stati applicati opportuni filtri che restituiscono le misure di interesse per due sole voci di costo assegnate agli agenti (costi per *cellulari* e *stipendio lordo base*) e per i due semestri.

		2011					
		S1			S2		
nome	cognome	budget_c	consuntivo_c	scostamento_c	budget_c	consuntivo_c	scostamento_c
Cellulare	Brennan	850	960	110	650	500	-150
	Bronsen	980	1.230	250	760	560	-200
	Jensen	850	880	30	740	580	-160
	Olliver	780	690	-90	670	530	-140
	Smith	600	355	-245	410	380	-30
	Westin	1.430	1.170	-260	1.210	890	-320
Stipendio lordo base	Brennan	10.500	11.160	660	10.500	8.060	-2.440
	Bronsen	9.800	10.170	370	9.800	6.660	-3.140
	Jensen	12.600	12.390	-210	12.600	9.040	-3.560
	Olliver	9.100	9.110	10	9.100	6.480	-2.620
	Smith	9.450	9.540	90	9.450	6.840	-2.610
	Westin	14.000	14.200	200	14.000	9.800	-4.200

Figura A.2: Report per i requisiti 1 e 7 per agente di vendita

Una volta definita la struttura del report, intesa come la modalità di visualiz-

zazione e distribuzione per righe e per colonne delle misure e delle dimensioni, si lascia all'utente la possibilità di compiere restrizioni sugli attributi dimensionali, ad esempio ottenendo l'output di Figura A.1 per tutti i trimestri dell'anno e per tutte le voci di costo (analogamente per l'esempio di Figura A.2). Sulla base dei requisiti di analisi numero 3 e 9 si genera il report di Figura A.3.

3	Importo totale del costo a budget per tutte le sedi
9	Importo totale del costo consuntivato per tutte le sedi

nome	2011		
	budget_c	consuntivo_c	scostamento_c
MADRID	419.000	371.770	-47.230
MILANO	386.640	339.245	-47.395
MONACO	365.840	335.060	-30.780
NEW YORK	651.910	537.455	-114.455
TORONTO	457.690	413.540	-44.150

Figura A.3: Report per i requisiti 3 e 9

In questo ultimo caso si mostrano gli importi di costo complessivi per ciascuna sede, senza distinguere la loro attribuzione per singola voce di costo.

nome	anno	mese	budget_c	consuntivo_c	scostamento_c
MILANO	2011	Gennaio	31.780	31.290	-490
		Febbraio	32.500	33.405	905
		Marzo	32.500	34.010	1.510
		Aprile	32.550	33.520	970
		Maggio	32.550	30.040	-2.510
		Giugno	34.550	30.370	-4.180
		Luglio	32.270	30.930	-1.340
		Agosto	28.150	24.570	-3.580
		Settembre	31.290	30.540	-750
		Ottobre	32.400	30.160	-2.240
		Novembre	32.450	30.410	-2.040
		Dicembre	33.650		-33.650

Figura A.4: Report di controllo direzionale

Preso in considerazione il modello di report direzionale mostrato in Figura 2.12, si riporta in Figura A.4 l'output che ne rispecchia le caratteristiche. Per ciascuna sede (in questo caso è applicato un filtro sulla sola sede di Milano) si presentano i valori delle misure distinti per mese e per anno. Come già osservato negli esempi precedenti, anche in questo caso è lasciata all'utente la possibilità di ottenere un'analisi complessiva delle informazioni senza l'applicazione di restrizioni sulla dimensione Sede. Si può quindi generare un output sugli importi di costo complessivi di tutte le sedi frazionati per ciascun mese di un certo anno.

Sulla base dei requisiti di analisi numero 4 e 10 si genera il report mostrato in Figura A.5.

4	Totale del ricavo di vendita preventivato per agente, per tipologia di ricavo, per mese (per anno)
10	Ammontare complessivo del fatturato di un agente, per trimestre e per semestre

anno	mese	Brennan			Smith		
		budget_r	consuntivo_r	scostamento_r	budget_r	consuntivo_r	scostamento_r
2011	Gennaio	25.800	10.500	-15.300	27.000	16.400	-10.600
	Febbraio	25.800	39.000	13.200	27.000	38.000	11.000
	Marzo	25.800	30.000	4.200	27.000	30.300	3.300
	Aprile	25.800	72.000	46.200	27.000	35.100	8.100
	Maggio	25.800	43.000	17.200	27.000	38.500	11.500
	Giugno	25.800	21.500	-4.300	27.000	35.500	8.500
	Luglio	25.800	20.000	-5.800	27.000	26.700	-300
	Agosto	25.800	8.300	-17.500	27.000	14.500	-12.500
	Settembre	25.800	44.000	18.200	27.000	43.000	16.000
	Ottobre	25.800	27.000	1.200	27.000	41.800	14.800
	Novembre	25.800	37.500	11.700	27.000	28.500	1.500
	Dicembre	25.800		-25.800	27.000		-27.000

Figura A.5: Report per i requisiti 4 e 10

Le misure a budget e consuntivo riportate rappresentano i valori complessivi dei ricavi derivanti dalla vendita di prodotti o servizi prestati per ciascun agente di vendita, frazionati sui singoli mesi dell'anno. Trattandosi di misure

complessive è implicito che queste sono riconducibili all'insieme di tutti i clienti seguiti da un singolo agente. Nello specifico esempio si riporta un confronto tra due soli agenti, ma che eventualmente può essere esteso per tutti gli agenti di vendita oppure, al contrario, limitarsi alla valutazione delle misure per un singolo individuo.

Se si considerano invece i requisiti di analisi numero 5 e 11, il report che ne scaturisce è rappresentato in Figura A.6.

5	Ammontare complessivo dell'importo ricavato a preventivo di un agente per trimestre e semestre per tutti i prodotti o servizi
11	Totale del ricavo di vendita fatturato da un agente per prodotto, per mese (anno)

anno	semestre	nome	Brennan		
			budget_r	consuntivo_r	scostamento_r
2011	S1	Monitor LCD HD	42.000	81.000	39.000
		Monitor TFT XA	15.000	21.000	6.000
		Office Suite Licence	7.800	11.500	3.700
		Server Lion LX	30.000	32.500	2.500
		Server Power TT	18.000	28.000	10.000
		Sugar EE Subscription	42.000	42.000	0
	S2	Monitor LCD HD	42.000	38.300	-5.700
		Monitor TFT XA	15.000	17.000	2.000
		Office Suite Licence	7.800	8.500	700
		Server Lion LX	30.000	28.000	-2.000
		Server Power TT	18.000	22.500	4.500
		Sugar EE Subscription	42.000	24.500	-17.500

Figura A.6: Report per i requisiti 5 e 11

Preso in considerazione il singolo agente e conseguentemente la totalità dei clienti associati a quest'ultimo, sono riportati gli importi complessivi distribuiti sui semestri per ciascun prodotto o servizio fornito.

Grafici per analisi prestazionali

In Figura A.7 si mostra il grafico relativo all'ammontare complessivo dei ricavi consuntivati derivanti dallo svolgimento delle attività degli agenti di vendita.

Grafico sulla distribuzione del fatturato tra le sedi aziendali

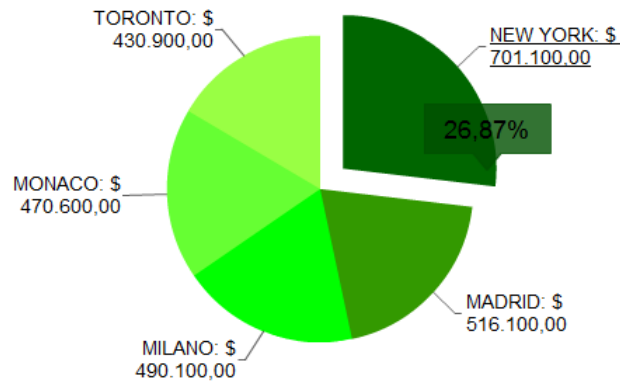


Figura A.7: Grafico di ripartizione dei ricavi totali a consuntivo per sede

```

SQL Query Area
1 SELECT UPPER(s.nome) as nome, sum(f.importo_co) as fatturato
2 FROM fact_ricavo f, sede s, agente a
3 WHERE f.id_agente=a.id_agente and a.id_sede=s.id_sede
4 GROUP BY s.id_sede
5 ORDER BY fatturato DESC;

```

Figura A.8: Query SQL per il grafico di Figura A.7

Spostando poi l'attenzione dalle sedi aziendali agli agenti di vendita, nel grafico a torta di Figura A.9 si rappresenta l'ammontare del costo a consuntivo che l'azienda sostiene per ogni singolo agente, mentre in Figura A.11 è mostrata la distribuzione del fatturato totale a consuntivo dell'azienda riconducibile a ciascun agente. Una rappresentazione di questo tipo permette di fare considerazioni aggiuntive sull'operato degli agenti. Ad esempio, dai dati mostrati nelle figure, è possibile dedurre che l'agente *Oliver* porta all'azienda il massimo utile (differenza tra i ricavi e i costi, pari a circa €200.000), grazie all'ottenimento del massimo livello di fatturato e all'abbattimento dei costi. Una diversa situazione invece è deducibile dalle informazioni riportate per l'agente *Bronsen*, il cui livello di utile è il minore fra tutti gli agenti (sotto la

soglia dei €20.000) nonostante quest'ultimo fatturi circa €410.000. In questa circostanza risulta evidente l'inefficienza dell'agente in questione.

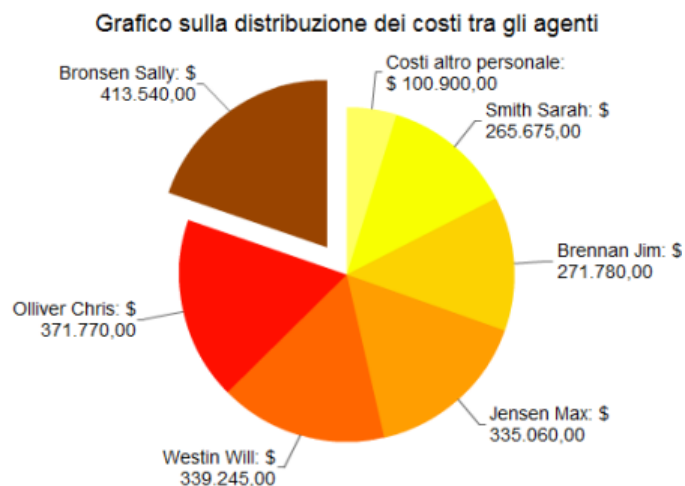


Figura A.9: Grafico di ripartizione dei costi a consuntivo per agente

```

SQL Query Area
1 SELECT CONCAT (a.cognome, ' ', a.nome) as cognome,
2 sum(f.importo_co) as costo
3 FROM fact_costo f, agente a
4 WHERE f.id_agente=a.id_agente
5 GROUP BY f.id_agente
6 ORDER BY costo;

```

Figura A.10: Query SQL per il grafico di Figura A.9

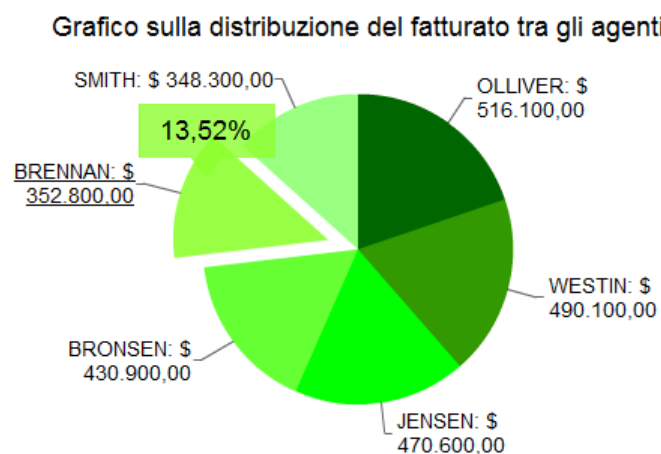


Figura A.11: Grafico di ripartizione dei ricavi a consuntivo per agente

```

SQL Query Area
1 SELECT UPPER(a.cognome) as cognome, sum(f.importo_co)
2 as fatturato
3 FROM fact_ricavo f, sede s, agente a
4 WHERE f.id_agente=a.id_agente and a.id_sede=s.id_sede
5 GROUP BY f.id_agente
6 ORDER BY fatturato DESC;

```

Figura A.12: Query SQL per il grafico di Figura A.11

Per gli altri agenti si nota invece un andamento standard delle loro attività con un fatturato complessivo medio dell'ordine dei €100.000 in cui i costi ed i ricavi sono ripartiti in modo proporzionale per ciascun soggetto.

Per quanto riguarda il rapporto sull'andamento mensile del fatturato degli agenti per l'anno 2011, la visualizzazione è mostrata nel grafico di Figura A.13. Questa tipologia di osservazione delle informazioni permette un confronto immediato dei valori ottenuti da ciascun agente sui singoli mesi di attività.

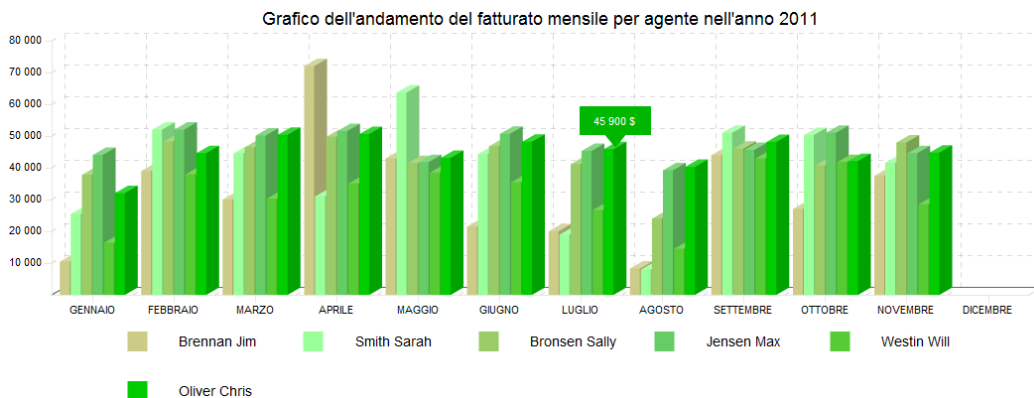


Figura A.13: Grafico di ripartizione mensile del fatturato per agente

```

SQL Query Area
1 SELECT a.cognome, sum (importo_co) as fatturato
2 FROM fact_ricavo f, time t, agente a
3 WHERE f.id_time=t.id_time and a.id_agente=f.id_agente
4 GROUP BY t.id_time, f.id_agente
5 ORDER BY t.id_time, a.cognome;

```

Figura A.14: Query SQL per il grafico di Figura A.13

Ad esempio si può osservare come l'agente *Oliver* mantenga pressoché co-

stante il livello di fatturato aziendale sulle diverse mensilità, a differenza dell'agente *Brennan* che invece presenta un andamento alternato, di cui i mesi di aprile ed agosto ne definiscono i livelli massimi e minimi. Questo particolare andamento può, ad esempio, essere sintomo di particolari richieste di prodotti o servizi da parte dei clienti dell'agente *Brennan* concentrati nel mese di aprile, oppure di una fidelizzazione di un nuovo cliente che ha deciso di rinnovare i propri beni tecnologici e a tale scopo si è rivolto all'agente a lui associato, o ancora si può pensare al mese di aprile come il periodo di scadenza delle licenze software fornite dall'azienda e che quindi necessitano un rinnovo.

La stessa rappresentazione è stata utilizzata per osservare, sempre nel dettaglio mensile, l'ammontare delle vendite dei singoli prodotti. In questo caso, come riportato in Figura A.15, compaiono sull'asse delle ascisse i prodotti ed i servizi forniti dall'azienda. Eseguendo opportune valutazioni incrociate con i risultati delle vendite degli agenti viste precedentemente si possono trarre conclusioni interessanti.

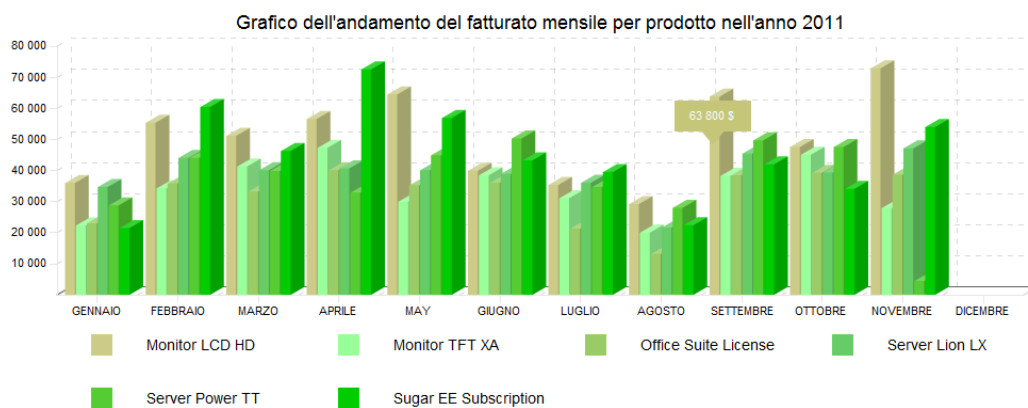


Figura A.15: Grafico di ripartizione mensile del fatturato per tipo di prodotto

```

SQL Query Area
1 SELECT p.nome, sum (importo_co) as fatturato
2 FROM fact_ricavo f, prodotto p, time t
3 WHERE f.id_time=t.id_time and p.id_prodotto=f.id_prodotto
4 GROUP BY t.id_time, p.id_prodotto
5 ORDER BY t.id_time, p.nome:

```

Figura A.16: Query SQL per il grafico di Figura A.15

Con riferimento all'esempio precedente, osservando il dettaglio del mese di aprile, si nota un elevato ammontare per le sottoscrizioni e rinnovi di licenze per il software indicato. Risulta quindi più realistica l'ipotesi che l'agente *Brennan*, nel mese indicato, ottenga il suo massimo livello di ricavo a causa dei rinnovi delle licenze che sono in scadenza per i clienti a lui associati.

Durante lo sviluppo dei documenti di reportistica si è presentata la possibilità di raccogliere le stesse informazioni rappresentandole sotto forma di incidenze percentuali, aggiungendo così valore alle già molteplici analisi e rapporti realizzati. In Figura A.17 si mostra un grafico rappresentante l'incidenza percentuale dei costi e dei ricavi di ciascun agente sul totale aziendale.

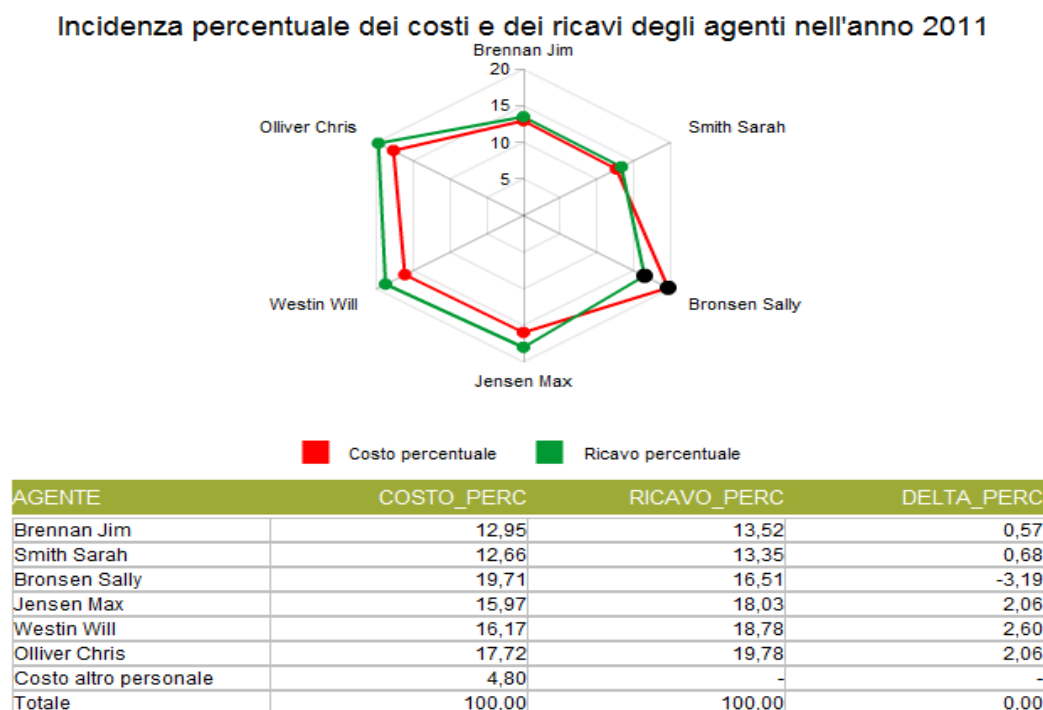


Figura A.17: Radar report per le incidenze percentuali

Il modello a radar utilizzato si presta per la consultazione semplice e intuitiva dei valori percentuali riportati, favorendo una veloce visione dei dati che costituiscono un'anomalia rispetto al loro andamento classico. Si nota ad esempio come l'agente *Bronsen* costituisce a livello percentuale un costo maggiore per l'impresa rispetto al ricavo, oppure come i valori degli agenti

Brennan e *Smith* siano pressochè identici rispetto a quelli registrati dagli altri agenti. I dati puntuali sono presentati anche in forma tabellare sulla griglia sottostante al grafico.

```

SQL Query Area
1 SELECT fr.agente as AGENTE, truncate(fc.costo,2) as COSTO_PERC, truncate(fr.ricavo,2) as RICAVO_PERC,
2      truncate ((fr.ricavo-fc.costo),2) as DELTA_PERC
3 FROM
4 (SELECT concat (a.cognome, ' ', a.nome) as agente, truncate (sum (f.importo_co)/x.tot*100,2) as costo
5 FROM
6 (SELECT sum(importo_co) as tot FROM fact_costo f) as x, fact_costo f, agente a
7 WHERE f.id_agente=a.id_agente
8 GROUP BY f.id_agente
9
10 UNION
11
12 SELECT 'Totale' as agente, truncate (sum (f.importo_co)/x.tot*100,2) as costo
13 FROM (SELECT sum(importo_co) as tot FROM fact_costo f) as x, fact_costo f) as fc LEFT JOIN
14
15 (SELECT concat (a.cognome, ' ', a.nome) as agente, truncate (sum (f.importo_co)/x.tot*100,2) as ricavo
16 FROM
17 (SELECT sum(importo_co) as tot FROM fact_ricavo f) as x, fact_ricavo f, agente a
18 WHERE f.id_agente=a.id_agente
19 GROUP BY f.id_agente
20
21 UNION
22
23 SELECT 'Totale' as agente, truncate (sum (f.importo_co)/x.tot*100,2) as ricavo
24 FROM (SELECT sum(importo_co) as tot FROM fact_ricavo f) as x, fact_ricavo f) as fr ON fc.agente=fr.agente

```

Figura A.18: Query SQL per il grafico di Figura A.17

Si mostra ora in Figura A.19 un ulteriore report, la cui struttura e rappresentazione grafica delle informazioni si discostano in parte dai modelli osservati in precedenza. Il fatturato complessivo, il corrispettivo valore a budget e la differenza rilevata per le varie sedi sono presentati nella ripartizione per agente di vendita sui singoli mesi. Gli scostamenti negativi vengono evidenziati con il colore rosso, in modo da catturare immediatamente l'attenzione del soggetto che riceve il rapporto. Al termine dell'elenco degli agenti viene presentato il totale del fatturato della sede di riferimento; informazione che non era presente nei rapporti basati sulle analisi dei requisiti mostrati nella sezione precedente.

La query SQL impiegata per reperire i dati all'interno del data warehouse è rappresentata in Figura A.20.

Questo report è stato costruito definendo un apposito file xaction, la cui sequenza d'azione effettua la connessione al data warehouse, esegue la query SQL e passa il result set di quest'ultima ad un file XML contenente la struttura grafica del report. I totali parziali dei valori a budget e consuntivo associati ai singoli agenti per ciascuna sede (identificabili sulla riga di colore verde nel rapporto) ed il totale complessivo a budget e consuntivo di ogni sede (identificabile sulla riga di colore azzurro nel rapporto) non vengono restituiti dalla query di Figura A.20 ma sono calcolati all'interno del file XML

direttamente al momento della visualizzazione della pagina.

Sede: NEW YORK

Agente	Mese	Consuntivo	Budget	Delta
Brennan Jim				
	Gennaio	10.500	25.800	-15.300
	Febbraio	39.000	25.800	13.200
	Marzo	30.000	25.800	4.200
	Aprile	72.000	25.800	46.200
	Maggio	43.000	25.800	17.200
	Giugno	21.500	25.800	-4.300
	Luglio	20.000	25.800	-5.800
	Agosto	8.300	25.800	-17.500
	Settembre	44.000	25.800	18.200
	Ottobre	27.000	25.800	1.200
	Novembre	37.500	25.800	11.700
	Dicembre		25.800	
	Total	352.800	309.600	69.000

Agente	Mese	Consuntivo	Budget	Delta
Smith Sarah				
	Gennaio	16.400	27.000	-10.600
	Febbraio	38.000	27.000	11.000
	Marzo	30.300	27.000	3.300
	Aprile	35.100	27.000	8.100
	Maggio	38.500	27.000	11.500
	Giugno	35.500	27.000	8.500
	Luglio	26.700	27.000	-300
	Agosto	14.500	27.000	-12.500
	Settembre	43.000	27.000	16.000
	Ottobre	41.800	27.000	14.800
	Novembre	28.500	27.000	1.500
	Dicembre		27.000	
	Total	348.300	324.000	51.300

NEW YORK Total	701.100	633.600	120.300
-----------------------	----------------	----------------	----------------

Figura A.19: Rapporto sul fatturato degli agenti per ogni sede

```

1 SELECT s.nome as SEDE, CONCAT (a.cognome, ' ', a.nome) as AGENTE, t.nomemese
2     as MESE, sum(importo_bg) as BUDGET, sum (importo_co) as CONSUNTIVO,
3     sum(importo_co)-sum(importo_bg) as DELTA
4 FROM fact_ricavo f, agente a, sede s, time t
5 WHERE f.id_agente=a.id_agente and a.id_sede=s.id_sede and f.id_time=t.id_time
6 GROUP BY s.id_sede, a.id_agente, t.nomemese
7 ORDER BY s.nome, a.cognome, t.id_time

```

Figura A.20: Query SQL per recuperare i dati del report di Figura A.19

Questa scelta implementativa però, non preclude il fatto che lo stesso risultato lo si possa ottenere sfruttando l'operatore WITH ROLLUP fornito da MySQL, inserendolo nella clausola GROUP BY (come mostrato in Figura A.21).

```

SQL Query Area
1 SELECT s.nome as SEDE, CONCAT (a.cognome, ' ', a.nome) as AGENTE,
2     IFNULL(t.nomemese, '') as MESE ,
3     sum(importo_bg) as BUDGET, sum (importo_co) as CONSUNTIVO,
4     sum(importo_co)-sum(importo_bg) as DELTA
5 FROM fact_ricavo f, agente a, sede s, time t
6 WHERE f.id_agente=a.id_agente and a.id_sede=s.id_sede and
7     f.id_time=t.id_time and t.anno=2011
8 GROUP BY s.id_sede, a.id_agente, t.nomemese WITH ROLLUP;

```

Figura A.21: Query SQL: prima alternativa alla query di Figura A.20

Da notare come l'utilizzo di detto operatore non permette l'impiego della clausola ORDER BY, in quanto in MySQL risultano mutuamente esclusivi. La scelta tra le due possibili varianti, ricaduta poi sull'utilizzo della query di Figura A.20, è dovuta al riscontro di alcune problematiche intercorse nell'output dell'interrogazione SQL contenente l'operatore WITH ROLLUP durante il calcolo dei totali parziali (come evidenziato dalle frecce nella Figura A.22).

SEDE	AGENTE	MESE	BUDGET	CONSUNTIVO	DELTA
NEW YORK	Brennan Jim	Gennaio	25800	10500	-15300
NEW YORK	Brennan Jim	Febbraio	25800	39000	13200
NEW YORK	Brennan Jim	Marzo	25800	30000	4200
NEW YORK	Brennan Jim	Aprile	25800	72000	46200
NEW YORK	Brennan Jim	Maggio	25800	43000	17200
NEW YORK	Brennan Jim	Giugno	25800	21500	-4300
NEW YORK	Brennan Jim	Luglio	25800	20000	-5800
NEW YORK	Brennan Jim	Agosto	25800	8300	-17500
NEW YORK	Brennan Jim	Settembre	25800	44000	18200
NEW YORK	Brennan Jim	Ottobre	25800	27000	1200
NEW YORK	Brennan Jim	Novembre	25800	37500	11700
NEW YORK	Brennan Jim	Dicembre	25800	NULL	NULL
NEW YORK	Brennan Jim	Dicembre	309600	352800	43200
NEW YORK	Smith Sarah	Gennaio	27000	16400	-10600
NEW YORK	Smith Sarah	Febbraio	27000	38000	11000
NEW YORK	Smith Sarah	Marzo	27000	30300	3300
NEW YORK	Smith Sarah	Aprile	27000	35100	8100
NEW YORK	Smith Sarah	Maggio	27000	38500	11500
NEW YORK	Smith Sarah	Giugno	27000	35500	8500
NEW YORK	Smith Sarah	Luglio	27000	26700	-300
NEW YORK	Smith Sarah	Agosto	27000	14500	-12500
NEW YORK	Smith Sarah	Settembre	27000	43000	16000
NEW YORK	Smith Sarah	Ottobre	27000	41800	14800
NEW YORK	Smith Sarah	Novembre	27000	28500	1500
NEW YORK	Smith Sarah	Dicembre	27000	NULL	NULL
NEW YORK	Smith Sarah	Dicembre	324000	348300	24300
NEW YORK	Smith Sarah	Dicembre	633600	701100	67500

Figura A.22: Errori nell'esecuzione della query di Figura A.21

Anche se i calcoli effettuati sulle misure di interesse risultano corretti, My-

SQL associa automaticamente sia ai totali parziali che a quello complessivo il valore dell'ultimo attributo trovato appartenente alle dimensioni di aggregazione. Pertanto, in sede di rappresentazione del rapporto, ciò costituisce motivo di errore nella visualizzazione.

Un'altra alternativa, che non ricade nell'errore appena discusso, prevede l'utilizzo di quattro distinte query unite tramite l'operatore UNION, come mostrato in Figura A.23.

```

SQL Query Area
1 (SELECT s.nome as SEDE, CONCAT (a.cognome, ' ', a.nome) as AGENTE,
2     t.nomemese as MESE ,sum(importo_bg) as BUDGET, sum (importo_co)
3     as CONSUNTIVO, sum(importo_co)-sum(importo_bg) as DELTA
4 FROM fact_ricavo f, agente a, sede s, time t
5 WHERE f.id_agente=a.id_Agente and a.id_sede=s.id_sede and
6     f.id_time=t.id_time and t.anno=2011
7 GROUP BY s.id_sede, a.id_agente, t.nomemese)
8
9 UNION
10
11 (SELECT s.nome as SEDE, CONCAT (a.cognome, ' ', a.nome) as AGENTE,
12     NULL as MESE ,sum(importo_bg) as BUDGET, sum (importo_co)
13     as CONSUNTIVO, sum(importo_co)-sum(importo_bg) as DELTA
14 FROM fact_ricavo f, agente a, sede s, time t
15 WHERE f.id_agente=a.id_Agente and a.id_sede=s.id_sede and
16     f.id_time=t.id_time and t.anno=2011
17 GROUP BY s.id_sede, a.id_agente)
18
19 UNION
20
21 (SELECT s.nome as SEDE, NULL as AGENTE, NULL as MESE ,
22     sum(importo_bg) as BUDGET, sum (importo_co) as CONSUNTIVO,
23     sum(importo_co)-sum(importo_bg) as DELTA
24 FROM fact_ricavo f, agente a, sede s, time t
25 WHERE f.id_agente=a.id_Agente and a.id_sede=s.id_sede and
26     f.id_time=t.id_time and t.anno=2011
27 GROUP BY s.id_sede)
28
29 UNION
30
31 (SELECT NULL as SEDE, NULL as AGENTE, NULL as MESE ,
32     sum(importo_bg) as BUDGET, sum (importo_co) as CONSUNTIVO,
33     sum(importo_co)-sum(importo_bg) as DELTA
34 FROM fact_ricavo f, agente a, sede s, time t
35 WHERE f.id_agente=a.id_Agente and a.id_sede=s.id_sede and
36     .id_time=t.id_time and t.anno=2011
37 ORDER BY s.id_sede, a.id_agente, t.nomemese);

```

Figura A.23: Query SQL: seconda alternativa alla query di Figura A.20

Anche quest'ultima possibilità è stata scartata poiché risulta costosa in termini di tempo di calcolo, in quanto per eseguire l'interrogazione sono necessarie quattro aggregazioni ciascuna ottenuta con una diversa GROUP BY.

Nel dettaglio, la prima delle quattro query calcola gli importi mensili a budget, a consuntivo ed il relativo scostamento per ciascun agente di ogni sede. Questo primo risultato parziale è mostrato in Figura A.24.

SEDE	AGENTE	MESE	BUDGET	CONSUNTIVO	DELTA
NEW YORK	Brennan Jim	Gennaio	25800	10500	-15300
NEW YORK	Brennan Jim	Febbraio	25800	39000	13200
NEW YORK	Brennan Jim	Marzo	25800	30000	4200
NEW YORK	Brennan Jim	Aprile	25800	72000	46200
NEW YORK	Brennan Jim	Maggio	25800	43000	17200
NEW YORK	Brennan Jim	Giugno	25800	21500	-4300
NEW YORK	Brennan Jim	Luglio	25800	20000	-5800
NEW YORK	Brennan Jim	Agosto	25800	8300	-17500
NEW YORK	Brennan Jim	Settembre	25800	44000	18200
NEW YORK	Brennan Jim	Ottobre	25800	27000	1200
NEW YORK	Brennan Jim	Novembre	25800	37500	11700
NEW YORK	Brennan Jim	Dicembre	25800	NULL	NULL
NEW YORK	Smith Sarah	Gennaio	27000	16400	-10600
NEW YORK	Smith Sarah	Febbraio	27000	38000	11000
NEW YORK	Smith Sarah	Marzo	27000	30300	3300
NEW YORK	Smith Sarah	Aprile	27000	35100	8100
NEW YORK	Smith Sarah	Maggio	27000	38500	11500
NEW YORK	Smith Sarah	Giugno	27000	35500	8500
NEW YORK	Smith Sarah	Luglio	27000	26700	-300
NEW YORK	Smith Sarah	Agosto	27000	14500	-12500
NEW YORK	Smith Sarah	Settembre	27000	43000	16000
NEW YORK	Smith Sarah	Ottobre	27000	41800	14800
NEW YORK	Smith Sarah	Novembre	27000	28500	1500
NEW YORK	Smith Sarah	Dicembre	27000	NULL	NULL

Figura A.24: Output della prima SELECT di Figura A.23

La seconda query è definita allo scopo di restituire i totali parziali di ogni agente, come mostrato nella Figura A.25.

SEDE	AGENTE	MESE	BUDGET	CONSUNTIVO	DELTA
NEW YORK	Brennan Jim	NULL	309600	352800	43200
NEW YORK	Smith Sarah	NULL	324000	348300	24300
MILANO	Westin Will	NULL	432000	490100	58100
TORONTO	Bronsen Sally	NULL	504000	430900	-73100
MONACO	Jensen Max	NULL	408000	470600	62600
MADRID	Olliver Chris	NULL	456000	516100	60100

Figura A.25: Output della seconda SELECT di Figura A.23

SEDE	AGENTE	MESE	BUDGET	CONSUNTIVO	DELTA
NEW YORK	NULL	NULL	633600	701100	67500
MILANO	NULL	NULL	432000	490100	58100
TORONTO	NULL	NULL	504000	430900	-73100
MONACO	NULL	NULL	408000	470600	62600
MADRID	NULL	NULL	456000	516100	60100

Figura A.26: Output della terza SELECT di Figura A.23

SEDE	AGENTE	MESE	BUDGET	CONSUNTIVO	DELTA
NULL	NULL	NULL	2433600	2608800	175200

Figura A.27: Output della quarta SELECT di Figura A.23

Successivamente la terza interrogazione calcola i valori totali di ogni sede, rappresentati invece in Figura A.26, mentre, per concludere, la quarta query calcola il totale complessivo per tutte le sedi. Il corrispettivo output conclusivo è mostrato in Figura A.27

Bibliografia

- [Adamson 98] C. Adamson, M. Venerable, *Data Warehouse Design Solutions*, Wiley, 1998.
- [Albano 09a] A. Albano, *Basi di dati di supporto alle decisioni (Decision Support Databases)*, Appunti delle lezioni del corso di Sistemi informatici direzionali, Università di Pisa, 2009.
- [Albano 05b] A. Albano, G. Ghelli, R. Orsini, *Fondamenti di basi di dati*, Zanichelli, Bologna, 2005.
- [Barocco 09] V. Barocco, *La gestione della forza vendita*, Franco Angeli, Milano, 2009.
- [Bouman 09] R. Bouman, J. V. Dongen, *Pentaho Solutions - Business Intelligence and Data Warehousing with Pentaho and MySQL*, Wiley, Indianapolis, Indiana, 2009.
- [Castellano 03] N. Castellano, *Controllo di gestione ed informazioni*, Giuffrè, Milano, 2003.
- [Ciambotti 10] M. Ciambotti, *Il budget commerciale e di produzione*, Appunti delle lezioni del corso di Programmazione e Controllo 1, Università di Urbino, 2010.
- [Cinquini 08] L. Cinquini, *Strumenti per l'analisi dei costi - Vol.1 - Fondamenti di cost accounting*, Giappichelli, Torino, 2008.
- [CWH 11] *Community Wiki Home - Pentaho Community*, «<http://wiki.pentaho.com/>», 25 marzo 2011.

-
- [Dalli 07] D. Dalli, M. Bartoli, *Organizzazione e gestione delle vendite*, Dispense per il corso di Marketing e Gestione delle Vendite (CLEA), Università di Pisa, 2007.
- [Kimball 02] R. Kimball, M. Ross, *Data Warehouse - La guida completa*, Hoepli Informatica, Milano, 2002.
- [Marchi 08] L. Marchi, S. Marasca, A. Riccaboni, *Il controllo di gestione*, Knowità, Arezzo, 2008.
- [Paolini 04] A. Paolini, *Il controllo di gestione nelle imprese*, Giuffrè, Milano, 2004.
- [SGR 11] *SugarCRM Documentation - CRM Open Source*,
«<http://www.sugarcrm.com/crm/>», 19 marzo 2011.

Ringraziamenti

MATTEO RINGRAZIA...

Tanti mesi di lavoro e studio passati con passione e dedizione al raggiungimento di un obiettivo, alla stesura di una tesi, a risolvere continui problemi tecnici di un progetto, ad alcune inversioni di rotta e cambi di programma, ad innumerevoli spostamenti da una città all'altra e poi, come succede in molti casi, la parte più "complicata" è sempre la stessa: dedicare un pensiero alle persone che mi hanno incoraggiato e sostenuto, non solo in quest'ultimo periodo, ma in tutti questi anni.

Comincio quindi col ringraziare enormemente i miei genitori, Michele e Lucia, i primi ad avermi creduto, i primi ad aver condiviso con me l'entusiasmo per un nuovo esame sostenuto che mi avrebbe avvicinato sempre di più al traguardo oppure la delusione per una prova mancata, i primi ad entusiasmarsi per i miei successi o ad avere parole di conforto nei momenti più delicati e difficili, non solo nello studio. Come dimenticare poi tutti i parenti più stretti, gli zii per i quali ogni occasione era buona per interessarsi agli sviluppi dei miei studi ed i cugini, per i quali nutro un affetto fraterno, che mi hanno dimostrato sempre un sincero interessamento nonché un sentito affetto. Un sincero grazie anche alla Vale che ha condiviso con me ogni singolo momento di questo lungo percorso. Mi ha supportato e sopportato, incoraggiato, spronato, ha condiviso con me le scelte giuste e sbagliate, ma soprattutto mi è sempre stata vicino. Vi ringrazio tutti di cuore.

Ringrazio il Prof. Albano, il mio relatore, per la disponibilità ad accettare questo incarico, per le ore dedicatemi nei colloqui di questi ultimi mesi, per

i consigli dispensati e per tutto il tempo speso affinché quest'ultimo lavoro fosse strutturato in modo tale da essere presentato nel migliore dei modi ed utilizzabile per future consultazioni e studi.

Ringrazio Stefano per tutto il tempo dedicatomi e le lunghe chiacchierate, ogni occasione era buona per insegnarmi qualcosa. Lo ringrazio soprattutto per la passione e l'impegno speso per trasmettermi competenze, capacità e conoscenze, non solo dal punto di vista professionale. Non tutti sono disposti a dedicare il proprio tempo agli altri (soprattutto se inesperti e alle prime armi), tra impegni incombenti e frequenti viaggi di lavoro. Questo lo apprezzo molto e gliene sono riconoscente.

Naturalmente questo ringraziamento va esteso anche a tutti i componenti della squadra. A partire dalle donne: Serena, Carla, Francesca e Rita, fino a Rolando, Massimiliano, Lauro, Diego e David, che oltre ad aver messo a disposizione la loro professionalità mi hanno accolto nel migliore dei modi. La cordialità, la disponibilità, le attenzioni ed in particolar modo la serenità di tutti, mi hanno in qualche modo facilitato nell'inserimento e per tutta la durata della mia permanenza in azienda. Ne avrò un gradito ricordo.

Ringrazio tutti gli amici di sempre. Come si addice ad un amico, anche loro hanno giocato un ruolo importante in questo lungo percorso, ciascuno a suo modo, certo, però anche questo ha contribuito senza ombra di dubbio al raggiungimento di un risultato per me molto importante.

Ringrazio gli amici dell'università, quelli che già conoscevo e quelli che ho conosciuto strada facendo. Li ringrazio per tutti i momenti spensierati passati tra una pausa e l'altra, per le lunghe maratone dedicate alla preparazione degli esami nelle aule studio, per aver reso piacevole la permanenza in facoltà, per aver condiviso con me le loro aspettative, le speranze e i desideri.

Quasi dimenticavo. Un grazie speciale a Marco che con me ha condiviso (in tutti i sensi) questa bella esperienza. Non saprò come sarebbe stata se avessi intrapreso da solo questo percorso, quello che so, però, è che oltre ad un ottimo collega, ho guadagnato un buon amico.

Grazie a tutti.

MARCO RINGRAZIA...

Impresa ardua trovare le parole quando si giunge al termine di un percorso così importante ed impegnativo, che mi ha visto protagonista nelle gioie e nei dispiaceri di tanti momenti che non dimenticherò facilmente. Tante cose sono cambiate da quel giorno quando varcai la soglia di questa facoltà per la prima volta e che hanno portato il sottoscritto ad intraprendere un percorso di maturazione e ad essere la persona che oggi sono e che ha tagliato questo traguardo.

Senza dubbio il primo ringraziamento va ai miei genitori per avermi permesso di raggiungere quest'obiettivo e soprattutto per non avermi mai angosciato in merito allo studio, anzi per avermi sempre sostenuto quando alcuni ostacoli sembravano insuperabili. Siete veramente fantastici.

A Martina, perché senza la mia sorellina non saprei chi prendere in giro, e perché alla fine sono solo dimostrazioni di affetto nei suoi confronti, che solo i migliori fratelli possono capire.

Un ringraziamento unico e speciale va a Silvia, non solo perché è stata fin dall'inizio e per tutta la durata di quest'avventura al mio fianco, sempre pronta ad incoraggiarmi con una buona parola, un abbraccio o un semplice sorriso nei momenti difficili, ma soprattutto perché senza di lei sarei perso nella vita di tutti i giorni. Grazie davvero.

Al resto della truppa della famiglia: zii, cugini, nonni e chi più ne ha più ne metta, a qualcuno che purtroppo se ne è andato prima di oggi, questo risultato è anche merito vostro.

Agli amici di sempre per tutti i momenti trascorsi assieme, per le risate senza motivo (che poi sono le migliori), per i consigli, per le lezioni di vita ricevute e perché sempre pronti a farmi dimenticare qualsiasi delusione.

Al mio datore di lavoro per gli innumerevoli permessi concessimi al fine di arrivare fin qui.

Un ringraziamento anche ai compagni di studio che in questi anni si sono avvicinati ed in particolare a Davide, Alfonso, Giorgio, Chiara, Nico-

la, Giovanni con cui ho trascorso momenti di sofferenza ma principalmente di spensieratezza, e specialmente a Matteo mio compagno in questa avventura finale, conosciuto grazie all'ambito universitario ma diventato oramai amico al di fuori di ogni contesto scolastico. Perché non è solo studio e conoscenza quello che l'università può darti, ma è soprattutto una maturità nell'affrontare la vita di tutti i giorni, nei rapporti sociali e nel rispetto verso il prossimo.

Già quest'ultima avventura, trascorsa in questi mesi con i colleghi della Bnova, a tutti loro va certamente un ringraziamento speciale per quello che mi hanno insegnato e per il tempo e la passione che mi hanno dedicato, in particolar maniera ringrazio Stefano che più che un tutor si è dimostrato un amico, sempre disponibile nonostante i numerosi impegni lavorativi, capace di affrontare qualsiasi problematica mantenendo la calma e il sorriso sulle labbra, ti sei dimostrato veramente un esempio da seguire. Non ti dimenticherò e terrò in considerazione i tuoi insegnamenti.

Questo risultato raggiunto al termine di questo percorso è sicuramente merito di tutti VOI, perché ognuno di VOI mi ha dato qualcosa di cui ho fatto tesoro in questi anni, e questo lavoro è il mio modo di dirvi GRAZIE.