Progettazione del Data Warehouse

Queste dispense sono state estratte dalle dispense originali del Prof. Stefano Rizzi, disponibili in http://www-db.deis.unibo.it/~srizzi/)
e sono state tratte dal libro di testo [Data Warehouse - teoria e pratica della Progettazione, Autori: Matteo Golfarelli, Stefano Rizzi, Editore: McGraw-Hill]
Esse quindi costituiscono anche un' indicazione degli argomenti del libro di testo svolti durante il corso.

Approccio top-down

- Analizza i bisogni globali dell' intera azienda e pianifica lo sviluppo del DW per poi progettarlo nella sua interezza
  - Promette ottimi risultati poiché si basa su una visione globale dell' obiettivo e garantisce in linea di principio di produrre un DW consistente e ben integrato
  - Il preventivo di costi onerosi a fronte di lunghi tempi di realizzazione scoraggia la direzione dall' intraprendere il progetto
  - Affrontare contemporaneamente l' analisi e la riconciliazione di tutte le sorgenti di interesse è estremamente complesso
  - Riuscire a prevedere a priori nel dettaglio le esigenze delle diverse aree aziendali impegnate è pressoché impossibile, e il processo di analisi rischia di subire una paralisi
  - Il fatto di non prevedere la consegna a breve termine di un prototipo non permette agli utenti di verificare l' utilità del progetto e ne fa scemare l' interesse e la fiducia

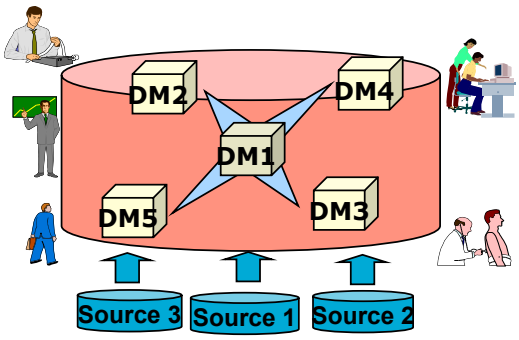
Approccio bottom-up

- Il DW viene costruito in modo incrementale, assemblando iterativamente più data mart, ciascuno dei quali incentrato su un insieme di fatti collegati a uno specifico settore aziendale e di interesse per una certa categoria di utenti
  - Determina risultati concreti in tempi brevi
  - Non richiede elevati investimenti finanziari
  - Permette di studiare solo le problematiche relative al data mart in oggetto
  - Fornisce alla dirigenza aziendale un riscontro immediato sull'effettiva utilità del sistema in via di realizzazione
  - Mantiene costantemente elevata l'attenzione sul progetto
  - Determina una visione parziale del dominio di interesse

Data Mart (DM)

DATA MART:

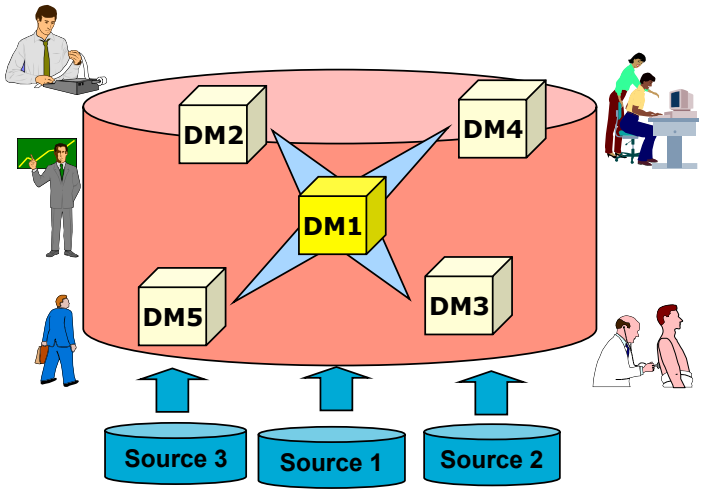
un sottoinsieme o un'aggregazione dei dati presenti nel DW primario, contenente l'insieme delle informazioni rilevanti per una particolare area del business, una particolare divisione dell'azienda, una particolare categoria di soggetti.



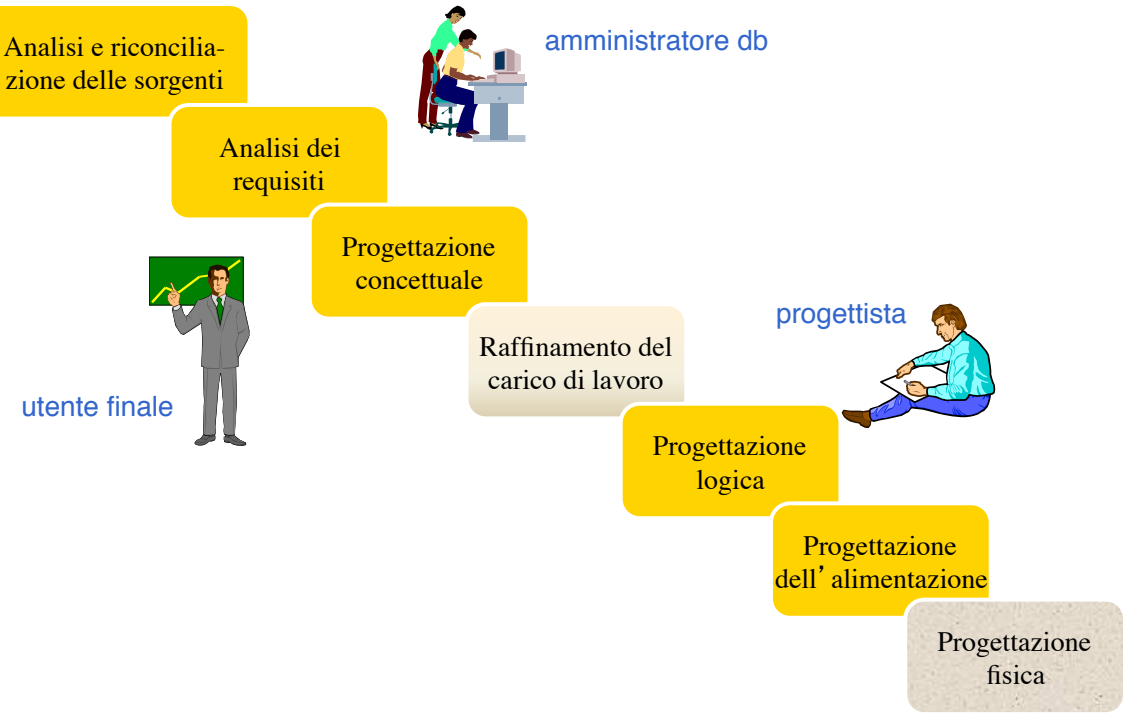
- ✓ Finchè consideriamo un DW costituito da un singolo DM, i due termini sono *sinonimi*
  - Questo è quanto accade in uno scenario di limitate dimensioni, come ad esempio nel caso di una unica source

Il primo data mart da prototipare...

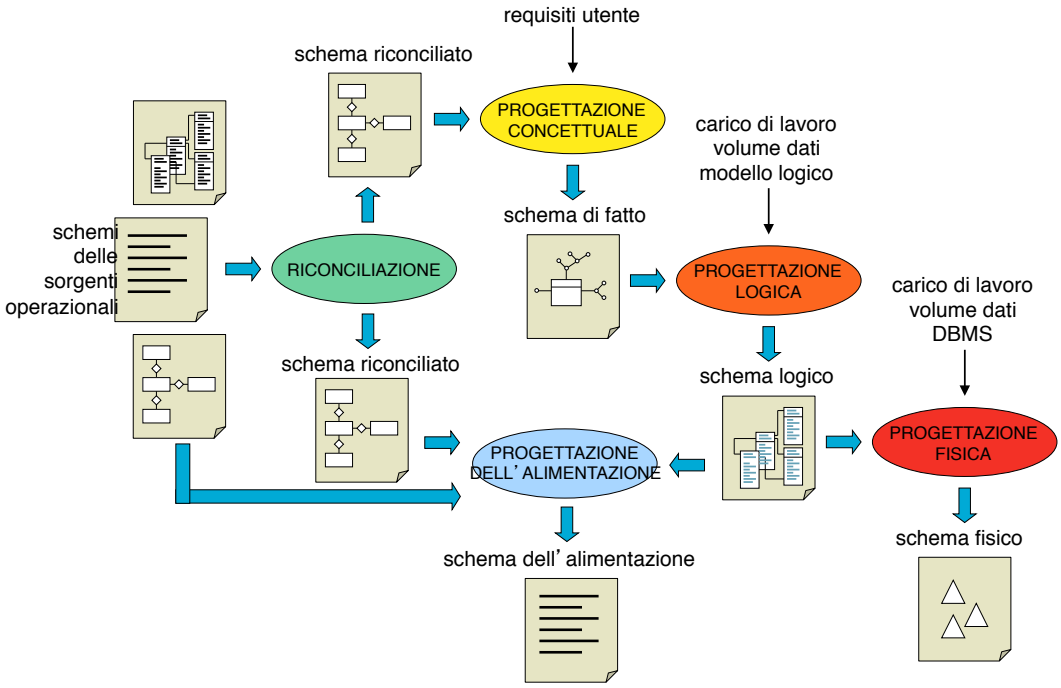
- ✓ deve essere quello che gioca il ruolo più strategico per l'azienda
- ✓ deve ricoprire un ruolo centrale e di riferimento per l'intero DW
- ✓ si deve appoggiare su fonti dati già disponibili e consistenti



La progettazione del data mart



La progettazione del data mart

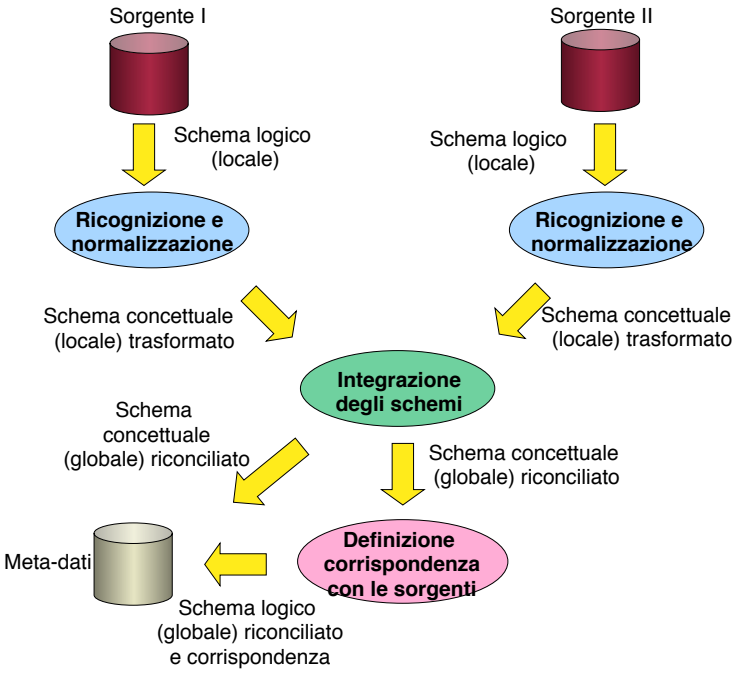


Analisi e riconciliazione delle sorgenti operazionali

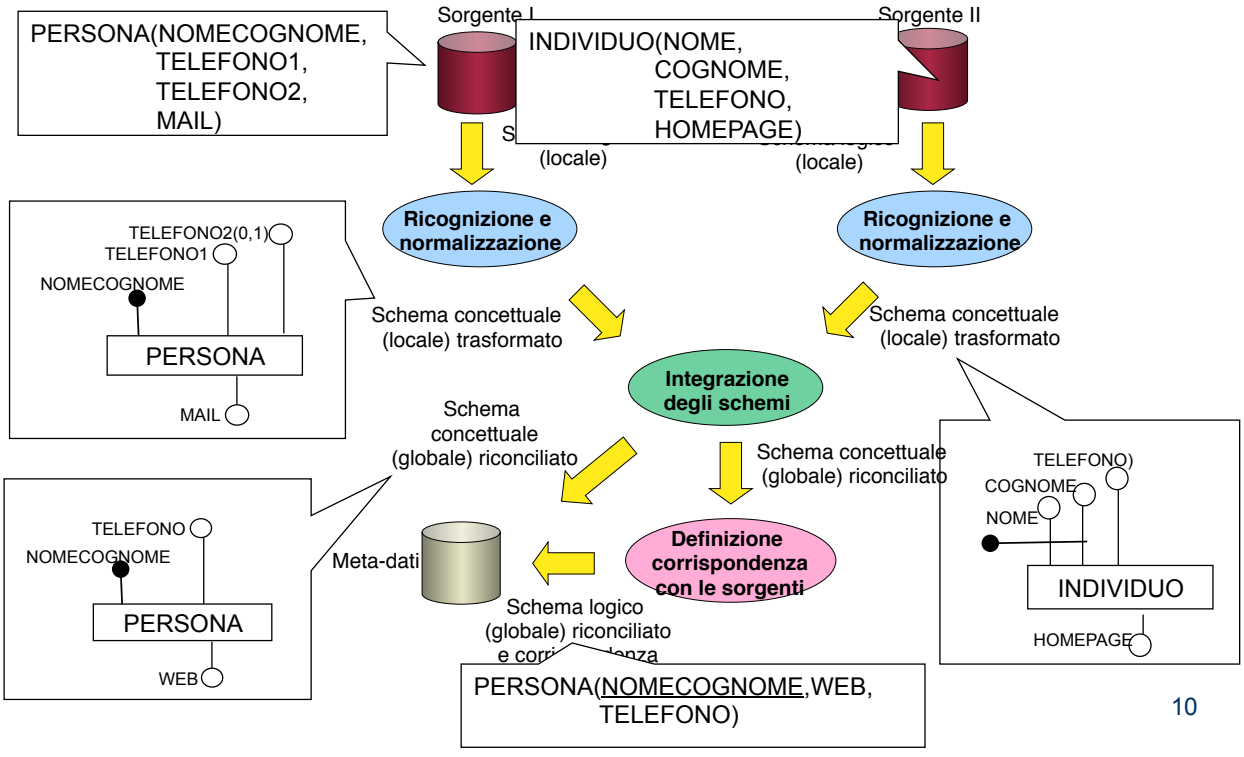
Rispetto alle dispense originali, la parte di integrazione di più sorgenti operazionali è stata eliminata in quanto verrà trattata più ampiamente nella parte del corso relativa alla **Integrazione delle Informazioni**

La parte sull'analisi di una singola sorgente è stata approfondita con varie considerazioni ed esempi.

Analisi e riconciliazione delle sorgenti operazionali



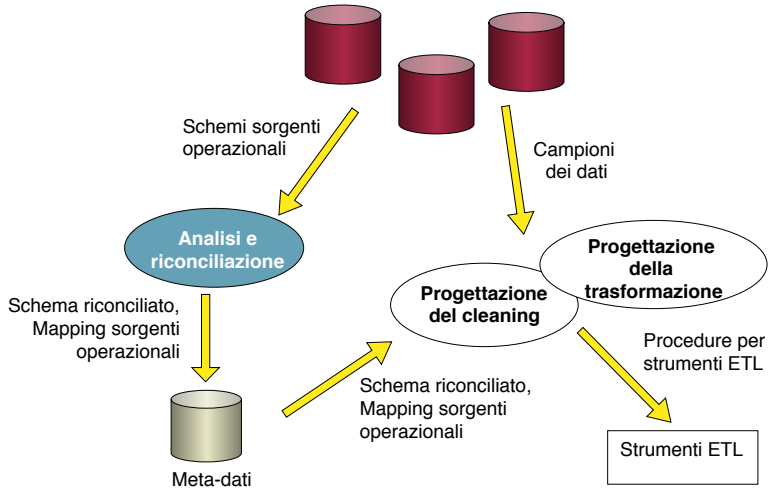
Analisi e riconciliazione delle sorgenti operazionali



CORRISPONDENZE (MAPPING)

SCHEMA LOGICO RICONCILIATO	SCHEMA LOGICO SORGENTE 1	SCHEMA LOGICO SORGENTE 2
	PERSONA	INDIVIDUO
NOMECOGNOME	NOMECOGNOME	NOME+ COGNOME
TELEFONO	TELEFONO1	TELEFONO
WEB	MAIL	HOMEPAGE

Progettazione del livello riconciliato



- ✓ La fase di integrazione è incentrata sulla componente intensionale delle sorgenti operazionali, ossia riguarda la consistenza degli schemi che le descrivono
- ✓ Pulizia e trasformazione dei dati operano a livello estensionale, ossia coinvolgono direttamente i dati veri e propri

Ricognizione e normalizzazione

- Il progettista, confrontandosi con gli esperti del dominio applicativo, acquisisce un' approfondita conoscenza delle sorgenti operazionali attraverso:
  - ✓ *Ricognizione* : esame approfondito degli schemi locali mirato alla piena comprensione del dominio applicativo;
  - ✓ *Normalizzazione*: l' obiettivo è correggere gli schemi locali per modellare in modo più accurato il dominio applicativo
- Ricognizione e normalizzazione devono essere svolte anche qualora sia presente una sola sorgente dati; qualora esistano più sorgenti, l' operazione dovrà essere ripetuta per ogni singolo schema
- Il punto di partenza, l' input del processo è costituito da:
  1. Schema logico e istanza della sorgente
  2. Eventuale schema concettuale "equivalente" allo schema logico
  3. Eventuale documentazione della sorgente operazionale

Progettazione di un DW da un DB relazionale

- Nel caso di DB relazionale (RDB) lo schema concettuale riconciliato è uno schema E/R che costituirà il punto di partenza per la progettazione concettuale del DW

- **Ricognizione e normalizzazione di un RDB**

  Oltre allo schema relazionale, più o meno completo con vincoli di integrità (chiavi, integrità referenziale, valori nulli, …), la documentazione generalmente disponibile per un DB relazionale può comprendere

  1. Uno schema E/R "equivalente" allo schema relazionale con eventualmente allegata una documentazione dello schema E/R
  2. Altra documentazione generica sul DB, quale le specifiche e requisiti del RDB e Manuali d' uso

- Se lo schema E/R del RDB non è disponibile esso viene ricavato attraverso tecniche di **Reverse engineering**

Documentazione di schemi E/R

- Uno schema E/R non è quasi mai sufficiente da solo a rappresentare tutti gli aspetti e vincoli di un dominio applicativo, per varie ragioni:
  1. in uno schema E/R compaiono solo i nomi dei vari concetti ma questo può essere insufficiente per comprenderne il significato.
  2. vari vincoli di integrità (proprietà dei dati rappresentati) non possono essere espressi direttamente dai costrutti del modello E/R
- **Documentazione di schemi E-R:** uno schema E/R è corredato con una documentazione di supporto che faciliti l'interpretazione dello schema stesso e a descrivere vincoli di integrità non esprimibili in E/R
- **Regole aziendali o business rules**
  - ✓ Una **descrizione** di un concetto (entità,associazione attributo) dello schema associazione del modello E-R (**Dizionario dei dati**)
  - ✓ Un **Vincolo di integrità**, sia esso la documentazione di un vincolo già nello schema E/R o la descrizione di un vincolo non esprimibile in E/R
  - ✓ Una **Derivazione** ovvero un concetto che può essere ottenuto attraverso un'inferenza o un calcolo da altri concetti dello schema (**Dato Derivato**)

Reverse engineering di schemi relazionali in schemi E/R

- Dallo schema relazionale ottenere lo schema E/R *equivalente*
- Procedimento inverso della **Progettazione logico-relazionale**: dato uno schema E/R tradurlo in uno schema relazionale
  - *Regole di semplificazione dello schema E/R* per eliminare identificatori esterni, gerarchie, …
  - *Regole di traduzione logico-relazionale* per tradurre uno schema E/R in uno schema relazionale normalizzato
- Il Reverse engineering di schemi relazionali in schemi E/R si basa sull'applicazione in senso inverso di queste regole di semplificazione e traduzione.
  Per rendere efficace il processo si deve partire da
  1. Uno schema relazionale completo con le indicazioni di chiavi e di integrità referenziale (chiavi esterne), valori nulli, …
  2. Uno schema relazionale normalizzato (infatti le regole *di traduzione logico-relazionale* forniscono uno schema relazionale normalizzato)

Individuazione di vincoli impliciti nei dati

- Analisi delle istanze per scoprire vincoli di integrità non noti ed impliciti nei dati. Da effettuare in ogni caso:
  1. **Schema E/R e documentazione disponibile**
     Nella realtà spesso non tutte le business rules sono documentate: ci sono vincoli di integrità non espressi nello schema E/R e non documentati .
  2. **Schema E/R non disponibile**
     Lo schema schema relazionale di partenza per effettuare Reverse engineering non è quasi mai completo e normalizzato
     - ✓ Individuazione di chiavi, chiavi esterne, …
     - ✓ Individuazione di dipendenze funzionali e normalizzazione dello schema

- Oltre a questi vincoli di integrità "classici", nel caso di progettazione di un DW vengono analizzati vincoli particolari quali la *convergenza*

Esempio

- RDB contenente la relazione

  PRODOTTO(CodProd, NomeProd, DescrizioneProd, DescrizioneCategoria, NomeCategoria, CodiceCategoria)

- Di tale RDB si può avere lo schema E/R corrispondente

- Si inizia quindi con l'analisi dei vincoli impliciti nei dati
  - Se lo schema E/R non c'è verrà ricavato dopo tale analisi

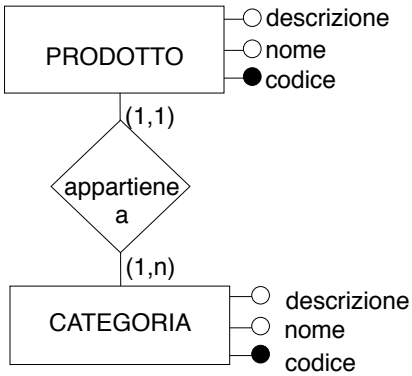
➢ Nelle slide che seguono viene data una intuizione di tale analisi, che verrà approfondita in seguito dove verranno introdotte le istruzioni SQL per poter verificare se i vincoli individuati sono validi o meno nell'istanza

Esempio

- Considerando i nomi significativi degli attributi e guardando alcune istanze si individuano i vincoli *ragionevoli*
  A. Il nome del prodotto è univoco? *SI!*
     `NomeProd` chiave alternativa
  B. Una categoria è identificata da CodiceCategoria ed ha un NomeCategoria ed una DescrizioneCategoria ? *SI!*
     `CodiceCategoria → NomeCategoria`
     `CodiceCategoria → DescrizioneCategoria`
  C. Un prodotto ha un' unica categoria ? *SI!*
     `CodProdotto → CodiceCategoria`
  D. Un prodotto ha sempre una categoria ? *SI!*
     `CodiceCategoria not null`
  E. Ad un categoria corrisponde un unico prodotto ? *NO!*
     `CodiceCategoria ↛ CodProdotto`

Esempio

- Supponendo validi i vincoli B, C, D, E si ottiene il seguente schema E/R riconciliato e normalizzato
- Lo schema viene completato con altri vincoli, sempre verificati sull' istanza (quale una categoria ha almeno un prodotto)



- Schema logico riconciliato e normalizzato
  `CATEGORIA(Codice,Nome,Descrizione) PRODOTTO (Codice,Nome,Descriz.,CodiceCategoria)`
  `FK: CodiceCategoria REF. CATEGORIA`
- In un' architettura a tre livelli, tale schema logico viene implementato in un nuovo RDB che verrà alimentato con i dati del RDB iniziale.

Vincoli di integrità *scoperti* sui dati

- Per definizione un vincolo di integrità deve essere valido per tutte le istanze dello schema
- Un vincolo di integrità *scoperto* sui dati vale ovviamente solo per l'istanza corrente
  - Riportare un vincolo scoperto sui dati sullo schema riconciliato deve essere una decisione del progettista!
- Come comportarsi se un vincolo viene considerato valido e riportato sullo schema riconciliato ma poi risulta non più verificato da alcune istanze del RDB?
  - Le istanze che non verificano il vincolo possono essere corrette oppure non riportate – in fase di alimentazione – nel RDB conciliato.

Analisi dei requisiti

- La fase di analisi dei requisiti ha l'obiettivo di raccogliere le esigenze di utilizzo del data mart espresse dai suoi utenti finali
- Essa ha un'importanza strategica poiché influenza le decisioni da prendere riguardo:
  - ✓ lo schema concettuale dei dati
  - ✓ il progetto dell'alimentazione
  - ✓ le specifiche delle applicazioni per l'analisi dei dati
  - ✓ l'architettura del sistema
  - ✓ il piano di avviamento e formazione
  - ✓ le linee guida per la manutenzione e l'evoluzione del sistema.

Fonti

- La “fonte” principale da cui attingere i requisiti sono i futuri utenti del data mart (*business users*)
  - ✓ La differenza nel linguaggio usato da progettisti e utenti, e la percezione spesso distorta che questi ultimi hanno del processo di warehousing, rendono il dialogo difficile e a volte infruttuoso

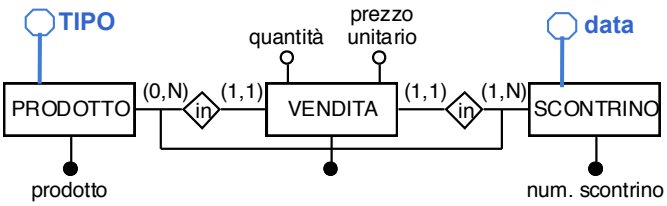
- Per gli aspetti più tecnici, saranno gli amministratori del sistema informativo e/o i responsabili del CED a fungere da riferimento per il progettista
  - ✓ In questo caso, i requisiti che dovranno essere catturati riguardano principalmente vincoli di varia natura imposti sul sistema di data warehousing

I fatti

- I fatti sono i concetti su cui gli utenti finali del data mart baseranno il processo decisionale; ogni fatto descrive una categoria di eventi che si verificano in azienda
  - ✓ Fissare le dimensioni di un fatto è importante poiché significa determinarne la granularità, ovvero il più fine livello di dettaglio a cui i dati saranno rappresentati. La scelta della granularità di un fatto nasce da un delicato compromesso tra due esigenze contrapposte: quella di raggiungere un’ elevata flessibilità d’ utilizzo e quella di conseguire buone prestazioni
  - ✓ Per ogni fatto occorre definire l’ intervallo di storicizzazione, ovvero l’arco temporale che gli eventi memorizzati dovranno coprire

ESEMPIO : vendite

- Schema E/R delle vendite:



1. Dimensioni = { prodotto, num.scontrino }

   ➔ massima granularità

   sono possibili analisi del numero clienti rispetto al tipo

2. Dimensioni = { prodotto, data}

   non sono possibili analisi del numero clienti rispetto al tipo

ESEMPIO : vendite (DB Operazionale)

- Scontrini:

Scontr12, del 02/02/02
ALIM_1, qty 10, prezzo 25
ALIM_2, qty 20, prezzo 12
…

Scontr13, del 02/02/02
ALIM_2, qty 24, prezzo 13
…

- **DB operazionale:**

PRODOTTO

PRODOTTO	TIPO
ALIM_1	Alimentare
ALIM_2	Alimentare

SCONTRINO

NUMERO	DATA
Scontr12	02/02/02
Scontr13	02/02/02

VENDITA

PRODOTTO	N_SCONTRINO	QUANTITA	PREZZO_UNITARIO
ALIM_1	Scontr12	12	25
ALIM_2	Scontr12	13	12
ALIM_2	Scontr13	24	13

ESEMPIO : vendite (DATA MART)

- Dimensioni = { prodotto, num.scontrino }

➔ **granularità transazionale** (massima granularità)
un evento primario nel Data Mart corrisponde ad una **sola istanza** del fatto nel DB Operazionale

FATTO VENDITA (Misure = { quantità, numero_clienti })

PRODOTTO	N_SCONTRINO	QUANTITA	NUMERO_CLIENTI
ALIM_1	Scontr12	12	1
ALIM_2	Scontr12	13	1
ALIM_2	Scontr13	24	1

ROLLUP

L'operatore di aggregazione per NUMERO_CLIENTI è COUNT DISTINCT di N_SCONTRINO

TIPO	N_SCONTRINO	QUANTITA	NUMERO_CLIENTI
Alimentare	Scontr12	25	1
Alimentare	Scontr13	24	1

L'operatore di aggregazione per QUANTITA è SUM

TIPO	QUANTITA	NUMERO_CLIENTI
Alimentare	59	2

ESEMPIO : vendite (DATA MART)

- Dimensioni = { prodotto, data }

➔ **granularità temporale**
un evento primario nel Data Mart corrisponde a **più istanze** del fatto nel DB Operazionale:
le misure del fatto devono essere calcolate tramite funzioni di aggregazione sulle istanze del DB operazionale
quantità = SUM(VENDITA.QUANTITA)
numero_clienti =COUNT(DISTINCT VENDITA.NSCONTRINO)

FATTO VENDITA (Misure = { quantità, numero_clienti })

PRODOTTO	DATA	QUANTITA	NUMERO_CLIENTI
ALIM_1	02/02/02	25	1
ALIM_2	02/02/02	24	2

ROLLUP

Non è più possibile valutare il NUMERO_CLIENTI rispetto al TIPO!

TIPO	DATA	QUANTITA	NUMERO_CLIENTI
Alimentare	02/02/02	59	*???*

I fatti

	Data mart	Fatti
commerciale/ manufatturiero	approvvigionamenti	acquisti, inventario di magazzino, distribuzione
	produzione	confezionamento, inventario, consegna, manifattura
	gestione domanda	vendite, fatturazione, ordini, spedizioni, reclami
	marketing	promozioni, fidelizzazione, campagne pubblicitarie
finanziario	bancario	conti correnti, bonifici, prestiti ipotecari, mutui
	investimenti	acquisto titoli, transazioni di borsa
	servizi	carte di credito, domiciliazioni bollette
sanitario	scheda di ricovero	ricoveri, dimissioni, interventi chirurgici, diagnosi
	pronto soccorso	accessi, esami, dimissioni
	medicina di base	scelte, revoche, prescrizioni
trasporti	merci	domanda, offerta, trasporti
	passeggeri	domanda, offerta, trasporti
	manutenzione	interventi
telecomunicazioni	traffico	traffico in rete, chiamate
	CRM	fidelizzazione, reclami, servizi
turismo	gestione domanda	biglietteria, noleggi auto, soggiorni
	CRM	frequent-flyers, reclami
gestionale	logistica	trasporti, scorte, movimentazione
	risorse umane	assunzioni, dimissioni, promozioni, incentivi
	budgeting	budget commerciale, budget di marketing
	infrastrutture	acquisti, opere

Glossario dei requisiti

Fatto	*Possibili dimensioni*	*Possibili misure*	*Storicità*
inventario di magazzino	prodotto, data, magazzino	quantità in magazzino	1 anno
vendite	prodotto, data, negozio	quantità venduta, importo, sconto	5 anni
linee d'ordine	prodotto, data, fornitore	quantità ordinata, importo, sconto	3 anni