

Il ciclo di vita del Data Warehouse

Prof. Stefano Rizzi

La progettazione del data mart

Analisi e riconcilia-
zione delle sorgenti

Analisi dei
requisiti

Progettazione
concettuale

Raffinamento del
carico di lavoro

Progettazione
logica

Progettazione
dell'alimentazione

Progettazione
fisica

amministratore db



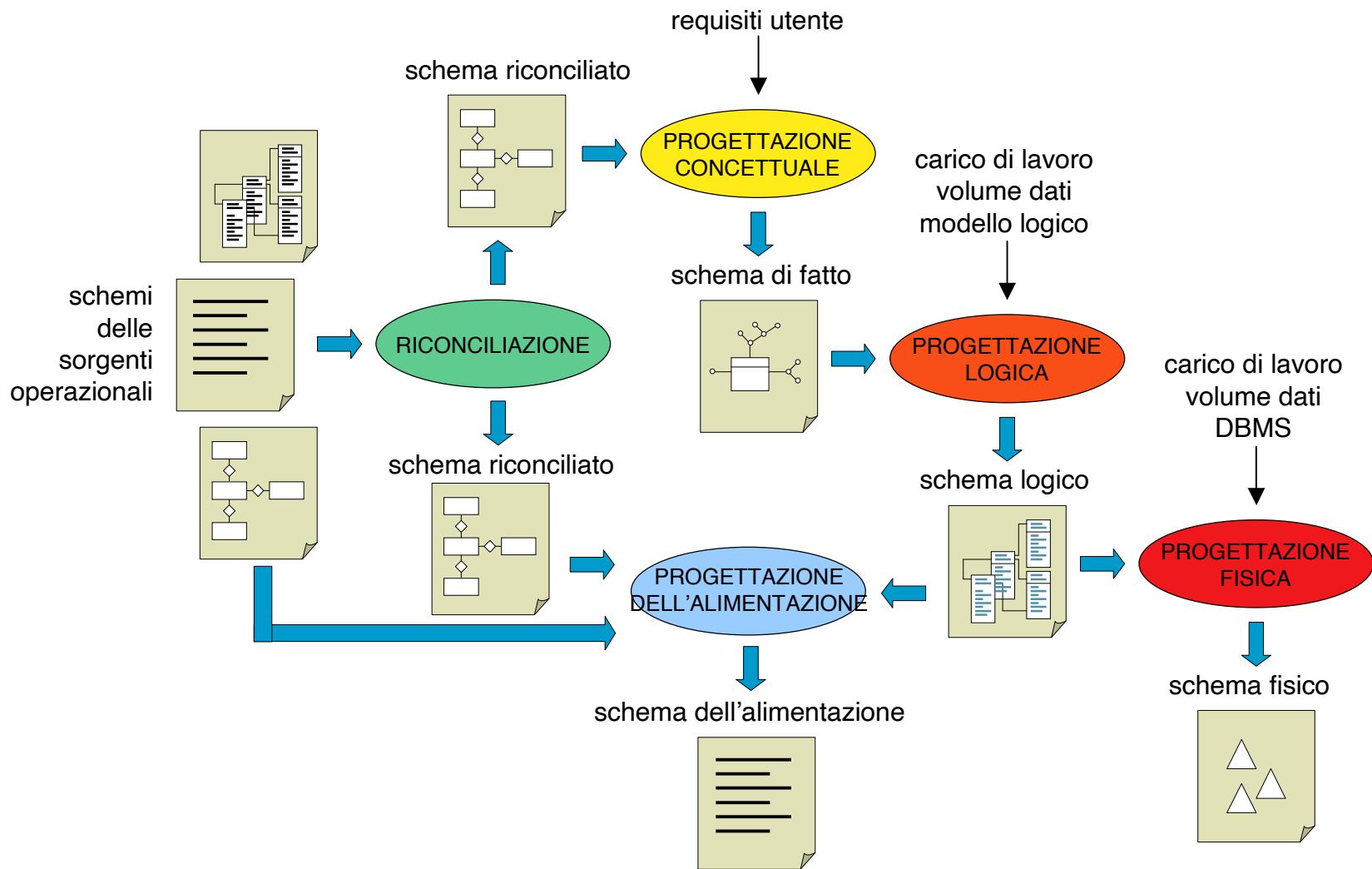
progettista



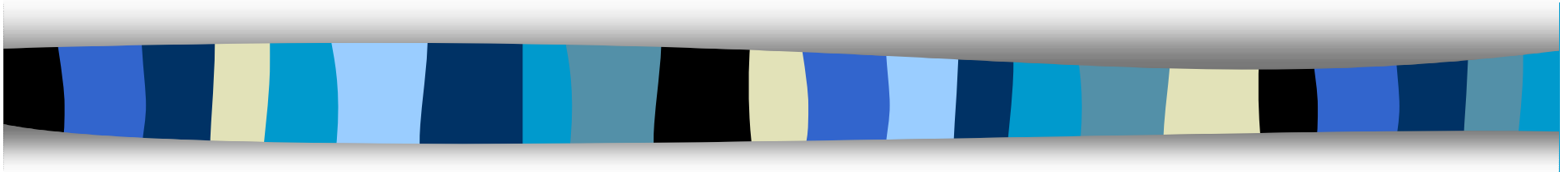
utente finale



La progettazione del data mart



Analisi dei requisiti



Prof. Stefano Rizzi



Obiettivi

- 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

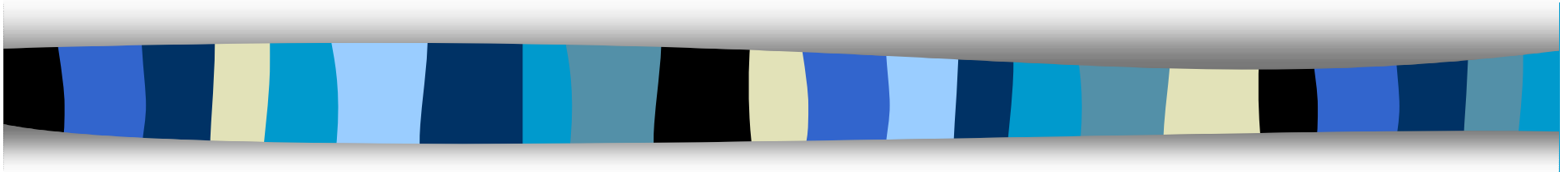




I fatti

- 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

Progettazione concettuale



Prof. Stefano Rizzi



Quale formalismo?

- Mentre è universalmente riconosciuto che un DW si appoggia sul modello multidimensionale, non c'è accordo sulla metodologia di progetto concettuale.
- Il modello Entity/Relationship è molto diffuso nelle imprese come formalismo per la documentazione dei sistemi informativi relazionali, ma *non può essere usato per modellare il DW*.



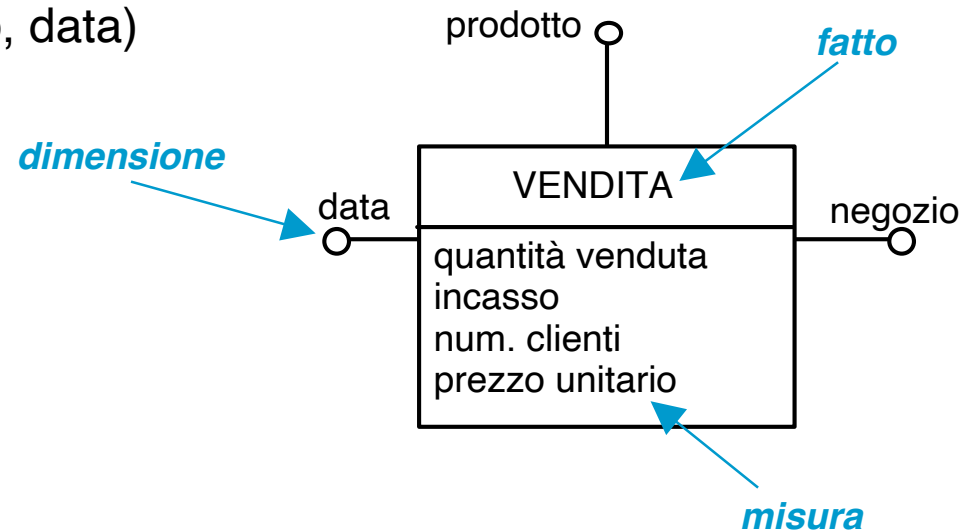
Il Dimensional Fact Model

- Il DFM è un modello concettuale grafico per data mart, pensato per:
 - ✓ supportare efficacemente il progetto concettuale;
 - ✓ creare un ambiente su cui formulare in modo intuitivo le interrogazioni dell'utente;
 - ✓ permettere il dialogo tra progettista e utente finale per raffinare le specifiche dei requisiti;
 - ✓ creare una piattaforma stabile da cui partire per il progetto logico (*indipendentemente dal modello logico target*);
 - ✓ restituire una documentazione a posteriori espressiva e non ambigua.
- La rappresentazione concettuale generata dal DFM consiste in un insieme di **scemi di fatto**. Gli elementi di base modellati dagli scemi di fatto sono i fatti, le misure, le dimensioni e le gerarchie

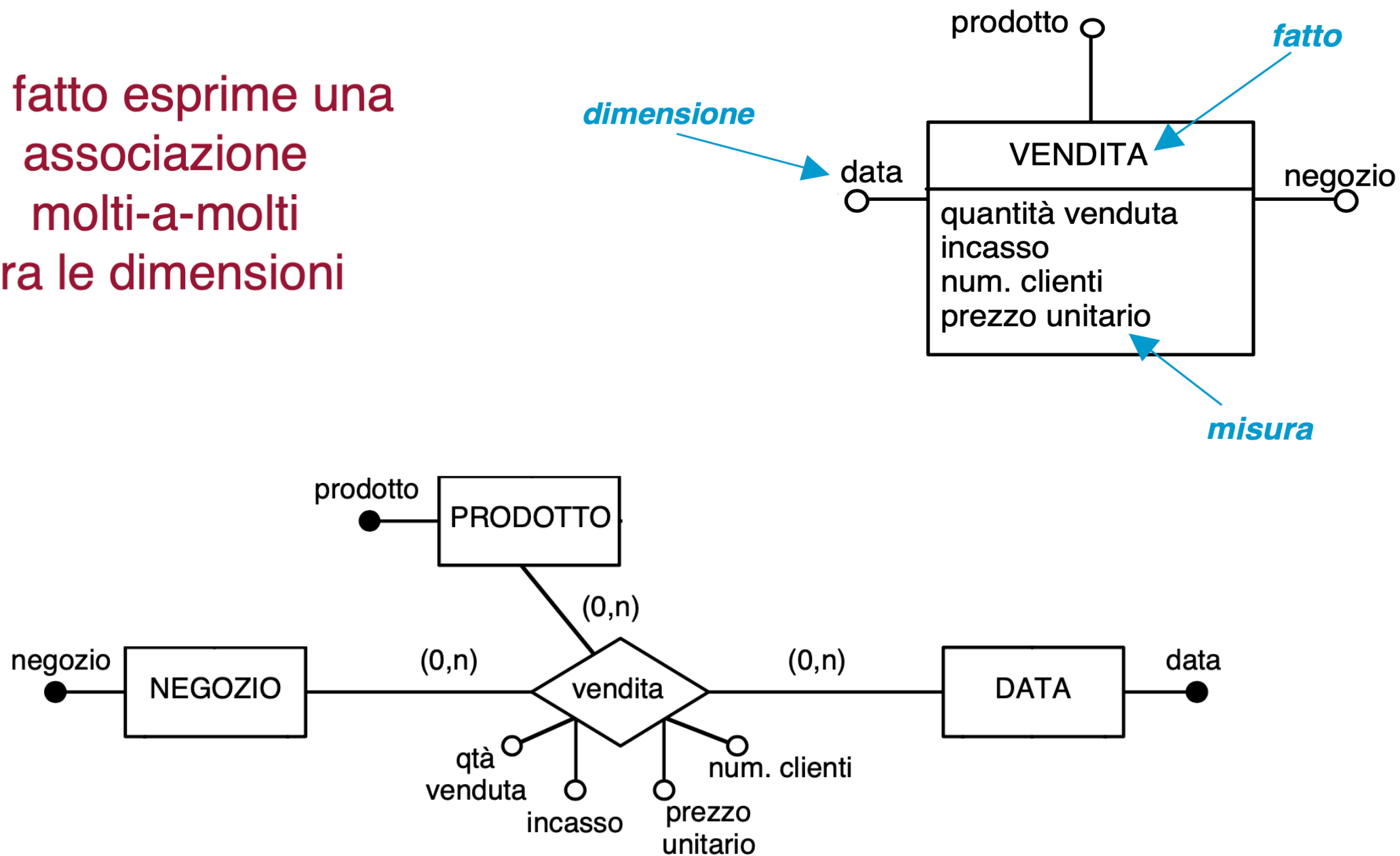
II DFM: costrutti di base

- Un *fatto* è un concetto di interesse per il processo decisionale; tipicamente modella un insieme di eventi che accadono nell'impresa (ad esempio: vendite, spedizioni, acquisti, ...). È essenziale che un fatto abbia aspetti dinamici, ovvero evolva nel tempo
- Una *misura* è una proprietà numerica di un fatto e ne descrive un aspetto quantitativo di interesse per l'analisi (ad esempio, ogni vendita è misurata dal suo incasso)
- Una *dimensione* è una proprietà con dominio finito di un fatto e ne descrive una coordinata di analisi (dimensioni tipiche per il fatto vendite sono prodotto, negozio, data)

Un fatto esprime una
associazione
multi-a-multi
tra le dimensioni

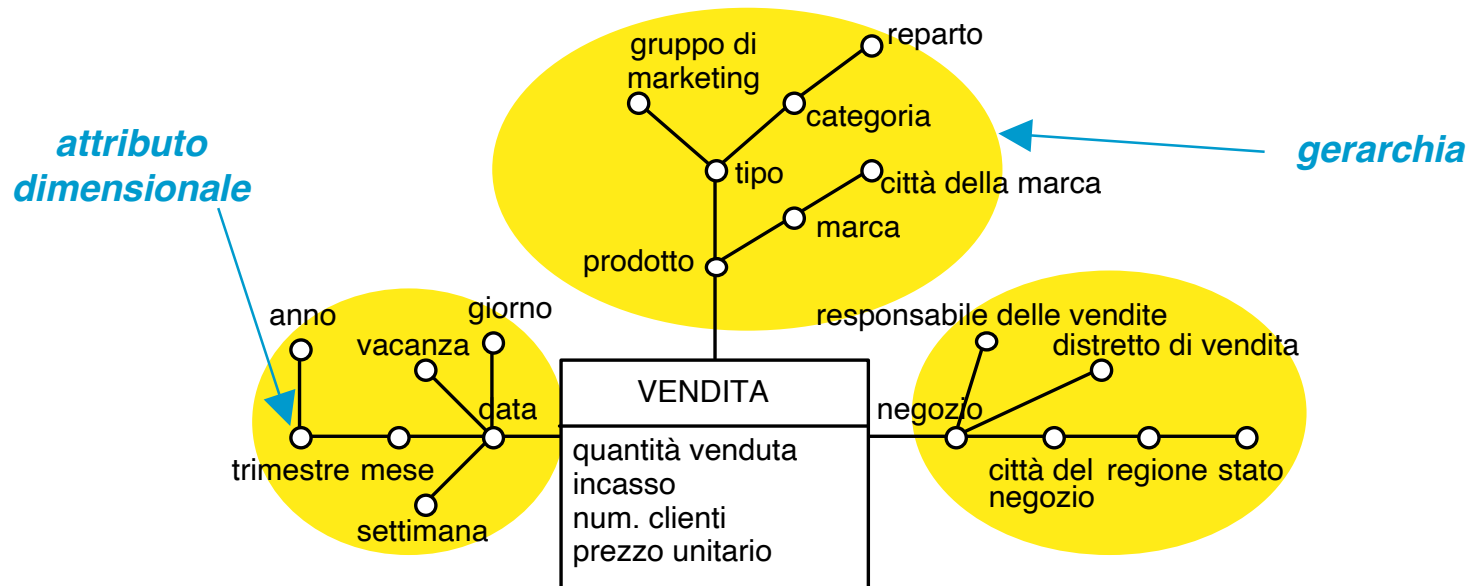


Un fatto esprime una
associazione
multi-a-molti
tra le dimensioni

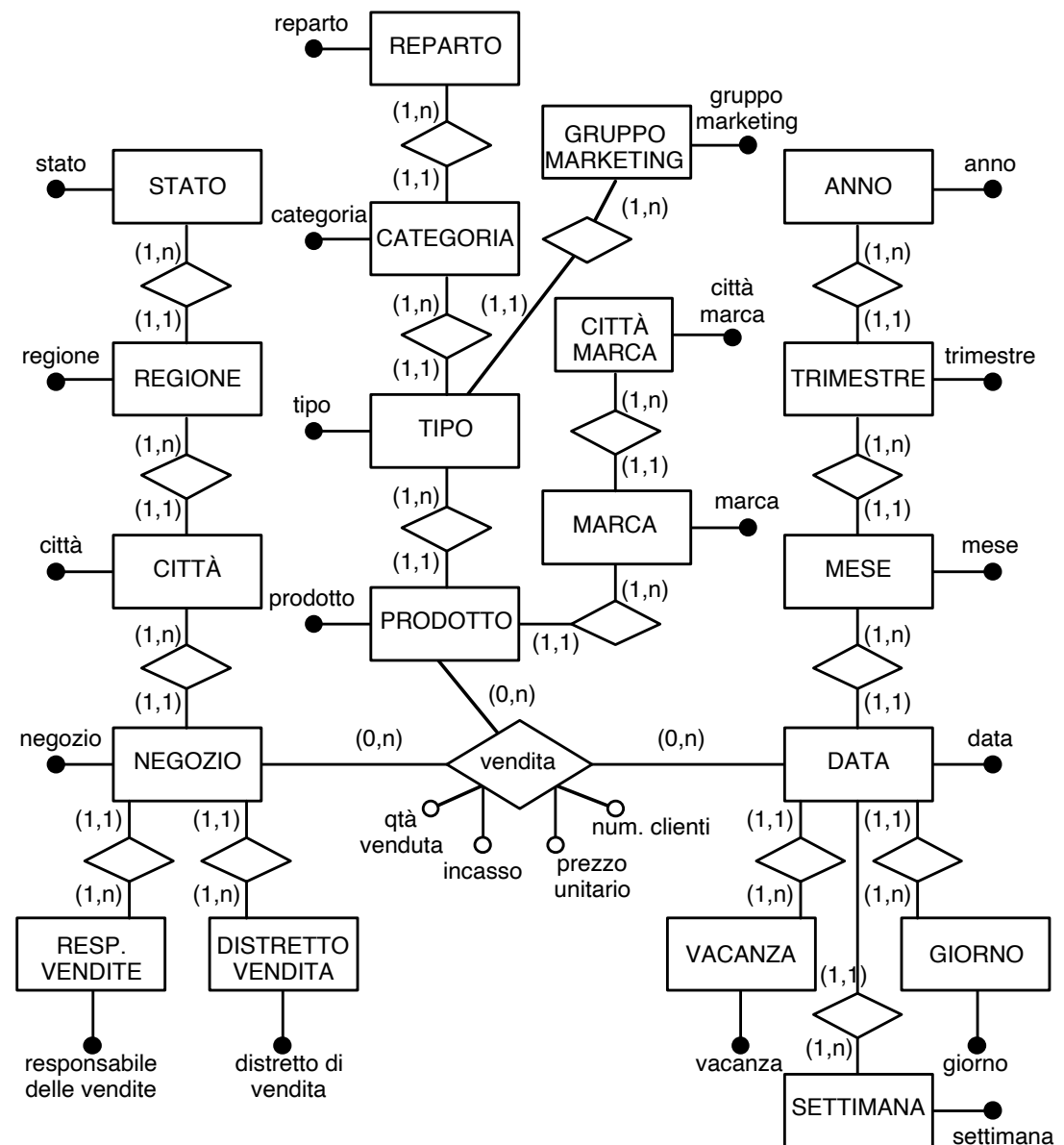


II DFM: costrutti di base

- Con il termine generale *attributi dimensionali* si intendono le dimensioni e gli eventuali altri attributi, sempre a valori discreti, che le descrivono (per esempio, un prodotto è descritto dal suo tipo, dalla categoria cui appartiene, dalla sua marca, dal reparto in cui è venduto)
- Una *gerarchia* è un albero direzionato i cui nodi sono attributi dimensionali e i cui archi modellano associazioni multi-a-uno tra coppie di attributi dimensionali. Essa racchiude una dimensione, posta alla radice dell'albero, e tutti gli attributi dimensionali che la descrivono



Il DFM: corrispondenza con l'E/R





“Naming conventions”

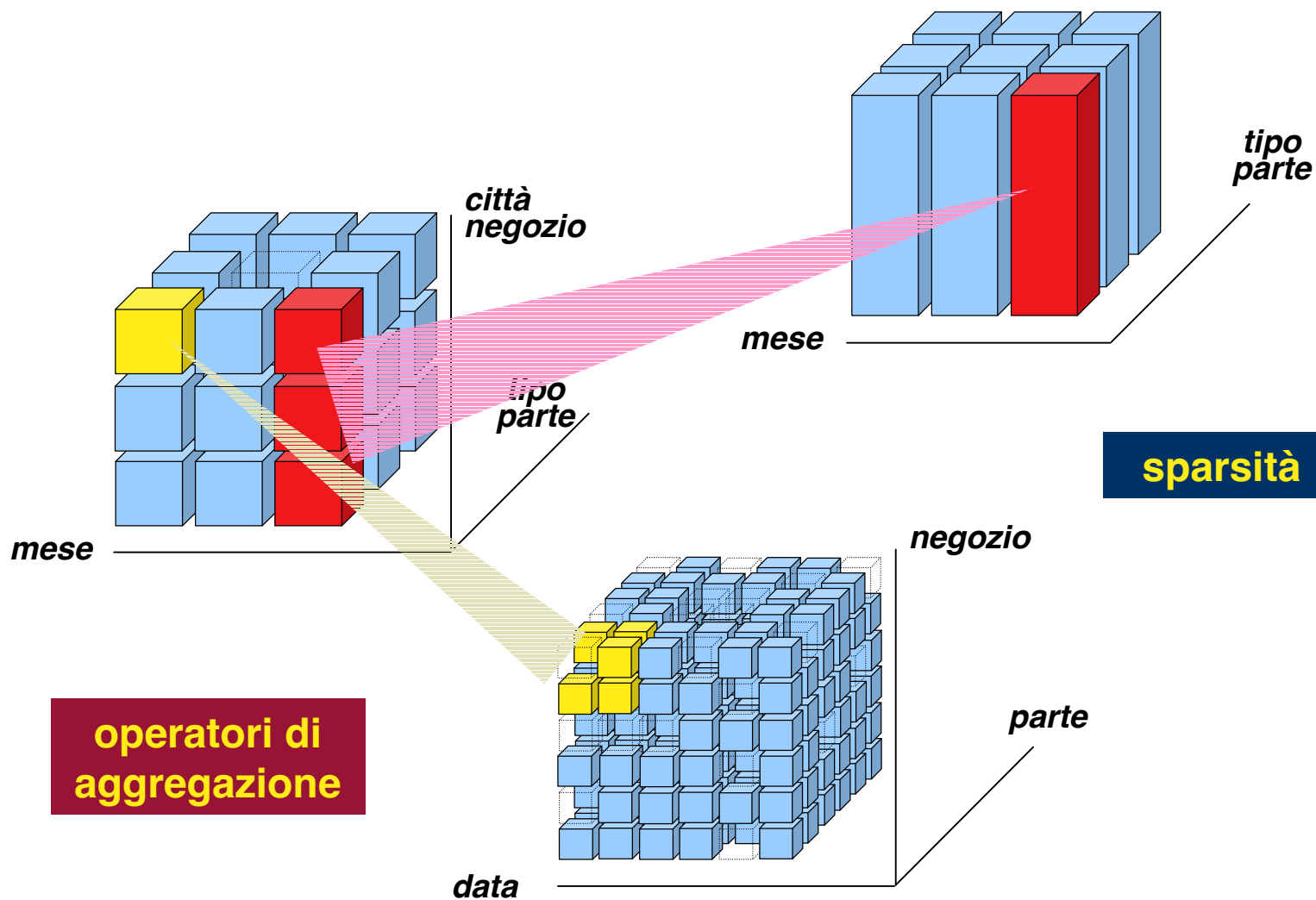
- Tutti gli attributi dimensionali in ciascuno schema di fatto devono avere nomi diversi
- Eventuali nomi uguali devono essere differenziati qualificandoli con il nome di un attributo dimensionale che li precede nella gerarchia
 - ✓ Ad esempio, *warehouse city* è la città in cui si trova un magazzino, mentre *store city* è la città in cui si trova un negozio
- I nomi degli attributi non dovrebbero riferirsi esplicitamente al fatto a cui appartengono
 - ✓ Ad esempio, si evitino *shipped product* e *shipment date*
- Attributi con lo stesso significato in schemi diversi devono avere lo stesso nome



Eventi e aggregazione

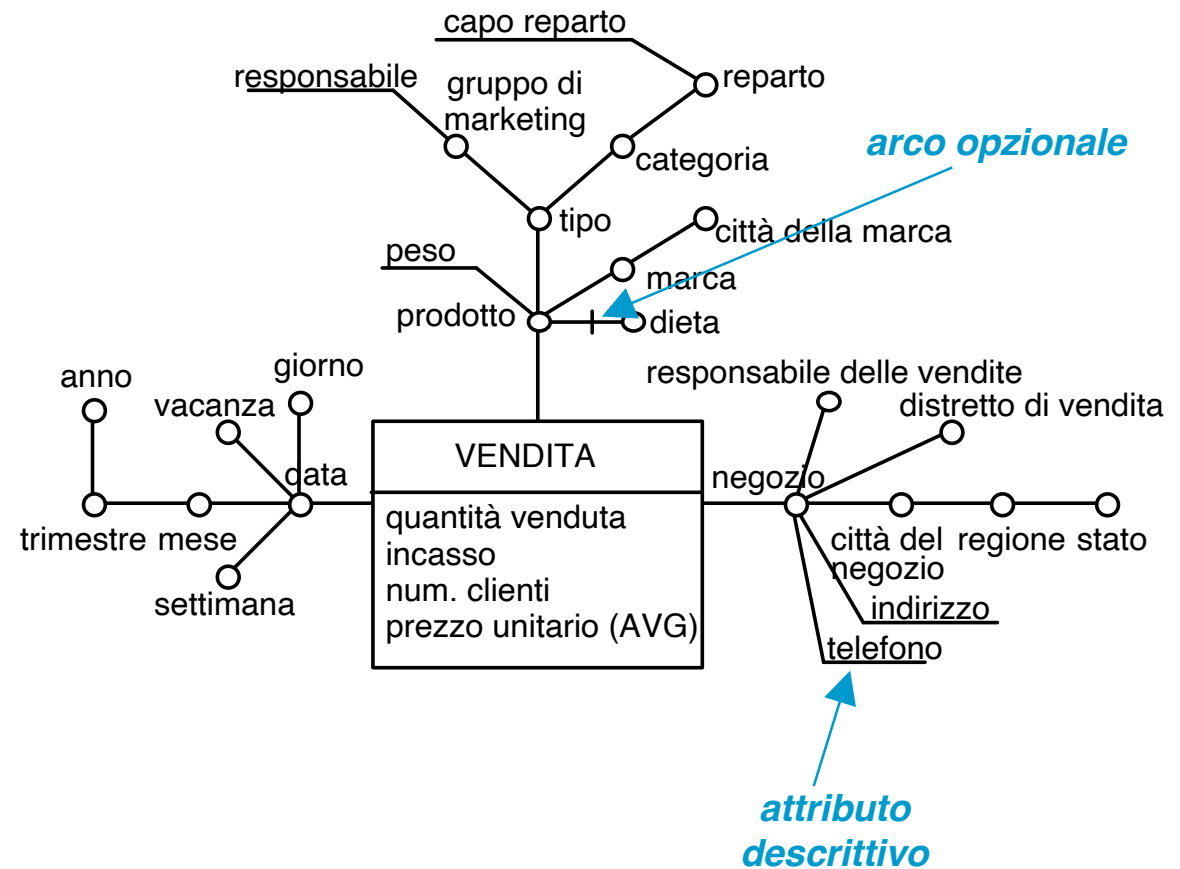
- Un **evento primario** è una particolare occorrenza di un fatto, individuata da una ennupla costituita da un valore per ciascuna dimensione. A ciascun evento primario è associato un valore per ciascuna misura
 - ✓ Con riferimento alle vendite, un possibile evento primario registra per esempio che, il 10/10/2001, nel negozio NonSoloPappa sono state vendute 10 confezioni di detersivo Brillo per un incasso complessivo pari a 25 euro
- Dato un insieme di attributi dimensionali (**pattern**), ciascuna ennupla di loro valori individua un **evento secondario** che aggrega tutti gli eventi primari corrispondenti. A ciascun evento secondario è associato un valore per ciascuna misura, che riassume in sé tutti i valori della stessa misura negli eventi primari corrispondenti
 - ✓ Pertanto, le gerarchie definiscono il modo in cui gli eventi primari possono essere aggregati e selezionati significativamente per il processo decisionale; mentre la dimensione in cui una gerarchia ha radice ne definisce la granularità più fine di aggregazione, agli altri attributi dimensionali corrispondono granularità via via crescenti

Eventi e aggregazione



II DFM: costrutti avanzati

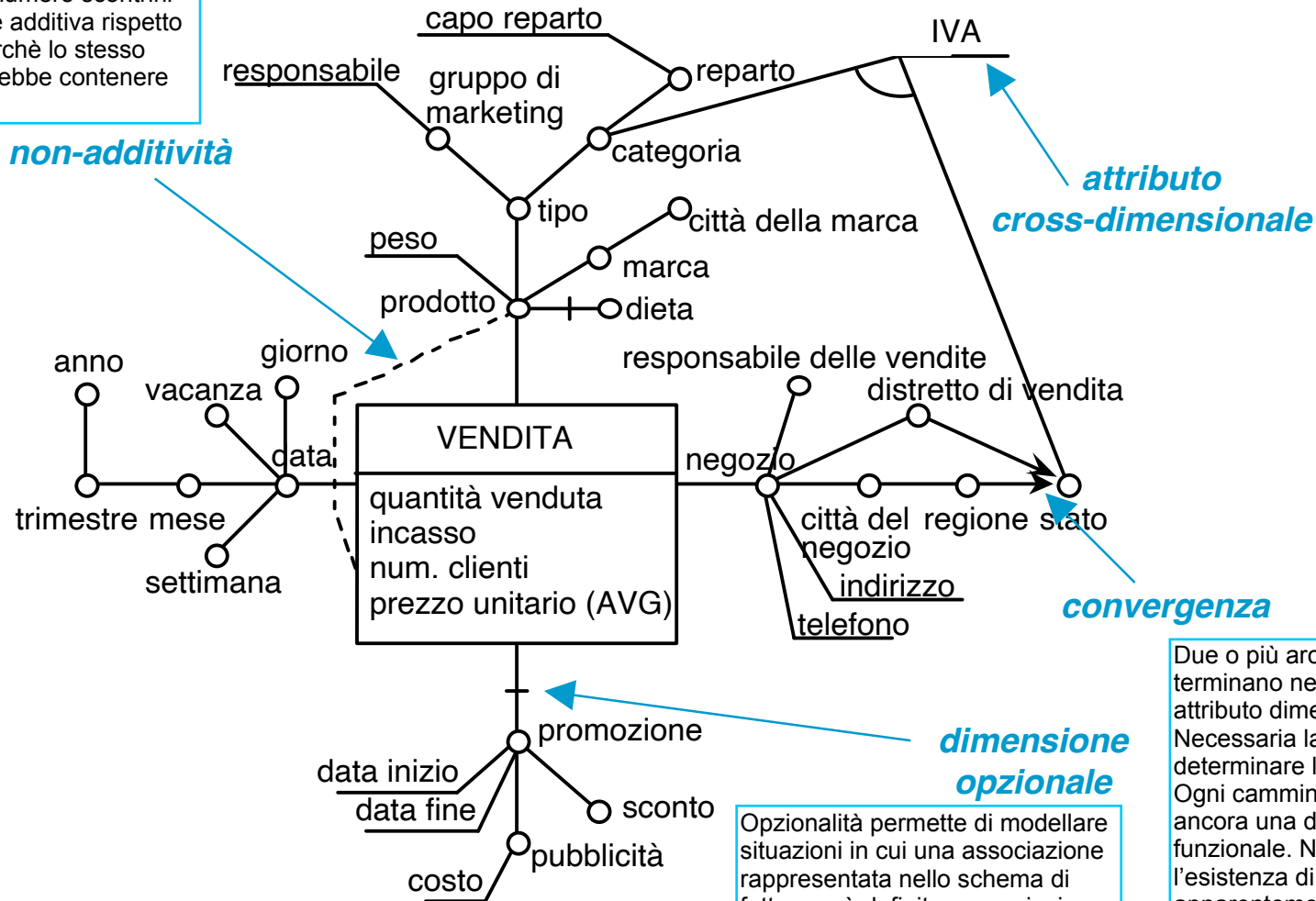
- Un *attributo descrittivo* contiene informazioni aggiuntive su un attributo dimensionale di una gerarchia, a cui è connesso da una associazione -a-uno. Non viene usato per l'aggregazione poiché ha valori continui e/o poiché deriva da un'associazione uno-a-uno
- Alcuni archi dello schema di fatto possono essere *opzionali*



II DFM: costrutti avanzati

Num.Clienti (numero scontrini emessi) non è additiva rispetto a prodotto perchè lo stesso scontrino potrebbe contenere più prodotti.

non-addittività

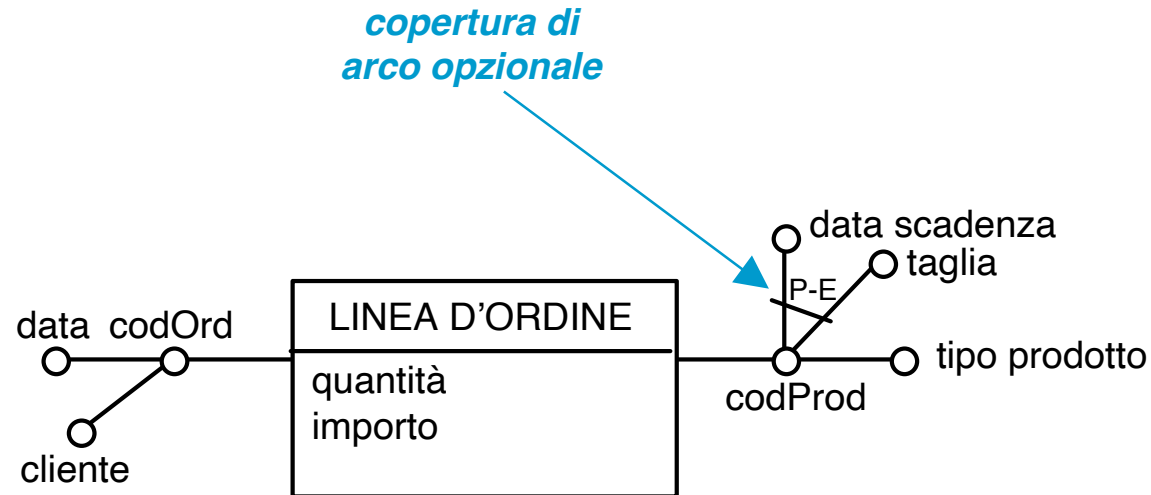


Addittività: una misura è detta additiva su una dimensione se i suoi valori possono essere aggregati lungo la corrispondente gerarchia tramite l'operatore somma, altrimenti è non-addittiva.

Opzionalità permette di modellare situazioni in cui una associazione rappresentata nello schema di fatto non è definita per un insieme di eventi. Dieta assume valore solo per i prodotti di tipo alimentare. Promozione opzionale significa che esisteranno alcuni eventi primari identificati solo da tre dimensioni

Due o più archi che terminano nello stesso attributo dimensionale. Necessaria la freccia per determinare la direzione. Ogni cammino rappresenta ancora una dipendenza funzionale. Non sempre l'esistenza di due attributi apparentemente uguali (città) determina una convergenza.

II DFM: costrutti avanzati



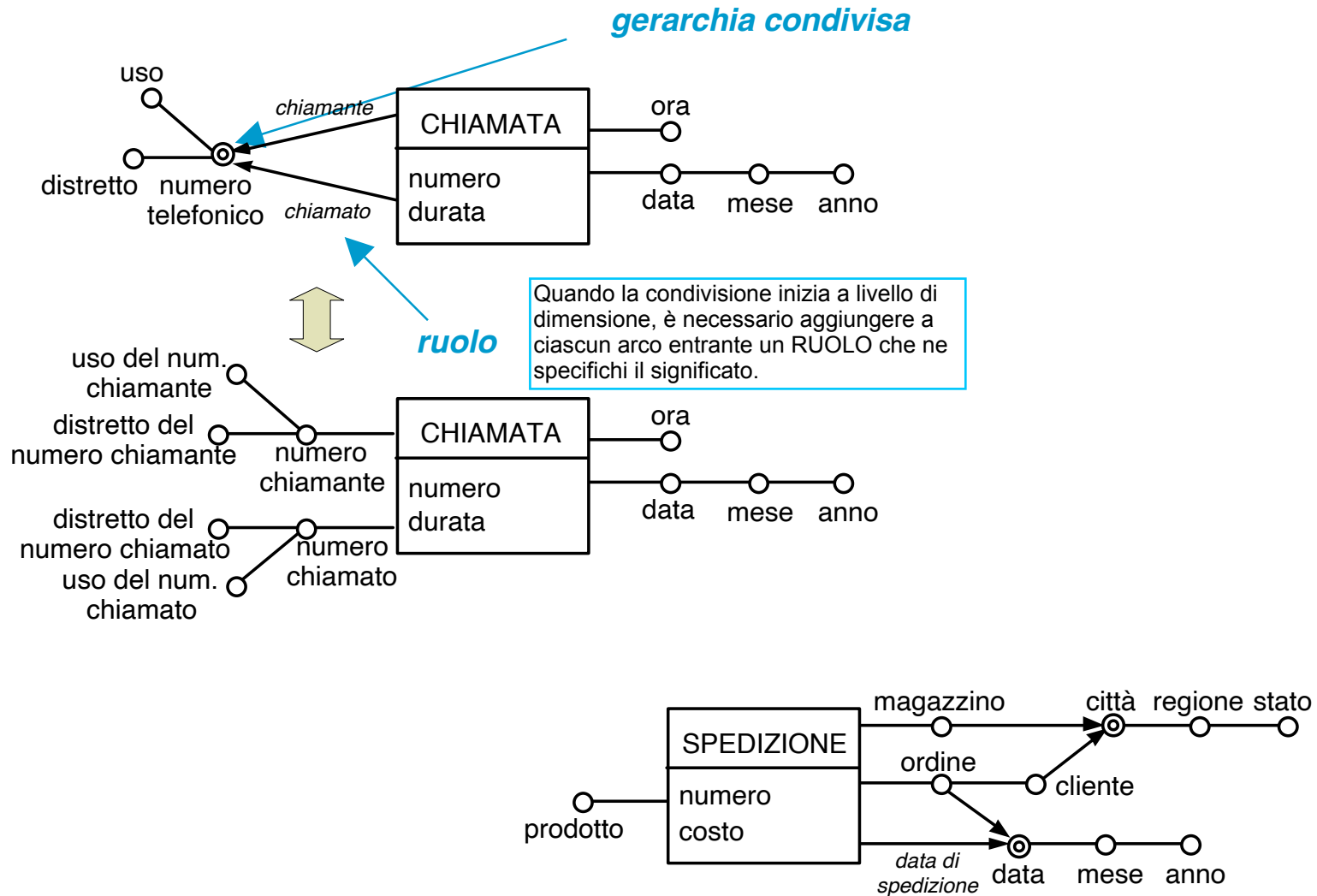
La copertura è totale se a ciascun valore dell'attributo "a" è sempre abbinato un valore di almeno uno dei figli; se invece esistono valori di "a" per i quali tutti i figli sono indefiniti, allora la copertura è detta parziale.

La copertura è esclusiva se in corrispondenza di ciascun valore di "a" si ha al massimo un valore per uno dei figli; se invece esistono valori di "a" abbinati a valori di due o più figli, la copertura è detta sovrapposta.

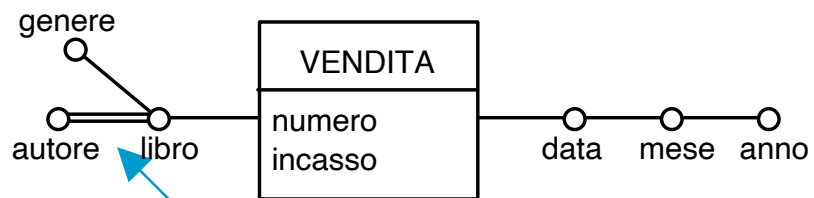
Quattro tipo di copertura: T-E, T-S, P-E e P-S.

Esempio: supponiamo che i prodotti possano essere di tre tipi: alimentari, abbigliamento e casalinghi. La data di scadenza e la taglia sono definiti solo per alimentari e abbigliamento rispettivamente, quindi la copertura è parziale ed esclusiva.

II DFM: costrutti avanzati



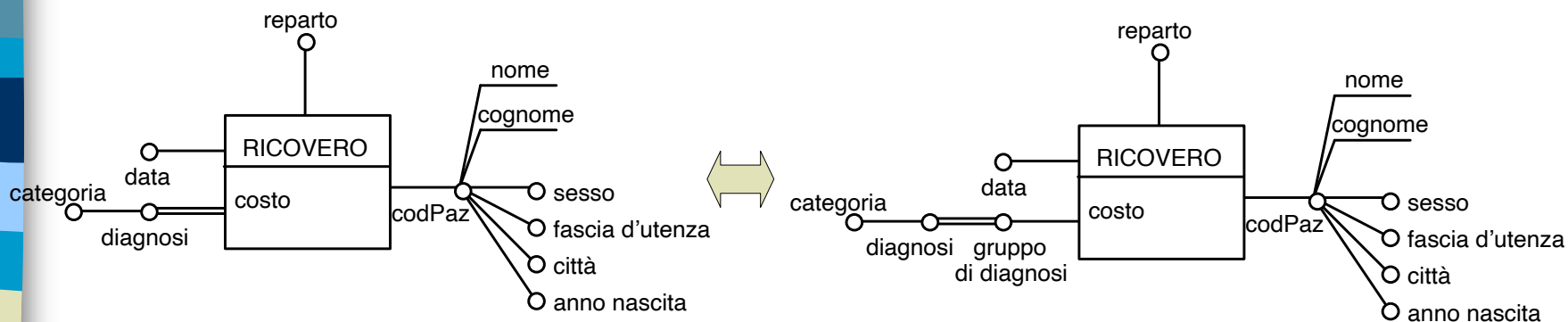
II DFM: costrutti avanzati



arco multiplo

Associazione molti a molti

Permette di includere attributi che, in corrispondenza di un singolo valore assunto dall'attributo padre, possono assumere valori multipli.



Additività

- L'aggregazione richiede di definire un operatore adatto per comporre i valori delle misure che caratterizzano gli eventi primari in valori da abbinare a ciascun evento secondario
- Da questo punto di vista è possibile distinguere tre categorie di misure:
 - ✓ **Misure di flusso**: si riferiscono a un periodo, al cui termine vengono valutate in modo cumulativo (il numero di prodotti venduti in un giorno, l'incasso mensile, il numero di nati in un anno)
 - ✓ **Misure di livello**: vengono valutate in particolari istanti di tempo (il numero di prodotti in inventario, il numero di abitanti di una città)
 - ✓ **Misure unitarie**: vengono valutate in particolari istanti di tempo, ma sono espresse in termini relativi (il prezzo unitario di un prodotto, la percentuale di sconto, il cambio di una valuta)

Misura di flusso: Quantità Venduta (di prodotti) è additiva -> La quantità venduta in un mese è la somma delle quantità vendute nei singoli giorni del mese

Misure di flusso

Misura di livello: Numero Prodotti (in inventario) NON è additiva, ma può essere aggregata con altri operatori. Esempio: numero minimo/massimo di prodotti

Misure di livello

Misure unitarie

Misura unitaria: Prezzo Unitario (di prodotti) NON è additiva, ma può essere aggregata con altri operatori. Esempio: media dei prezzi unitari di più prodotti

Gerarchie temporali

SUM, AVG, MIN, MAX

AVG, MIN, MAX

AVG, MIN, MAX

Gerarchie non temporali

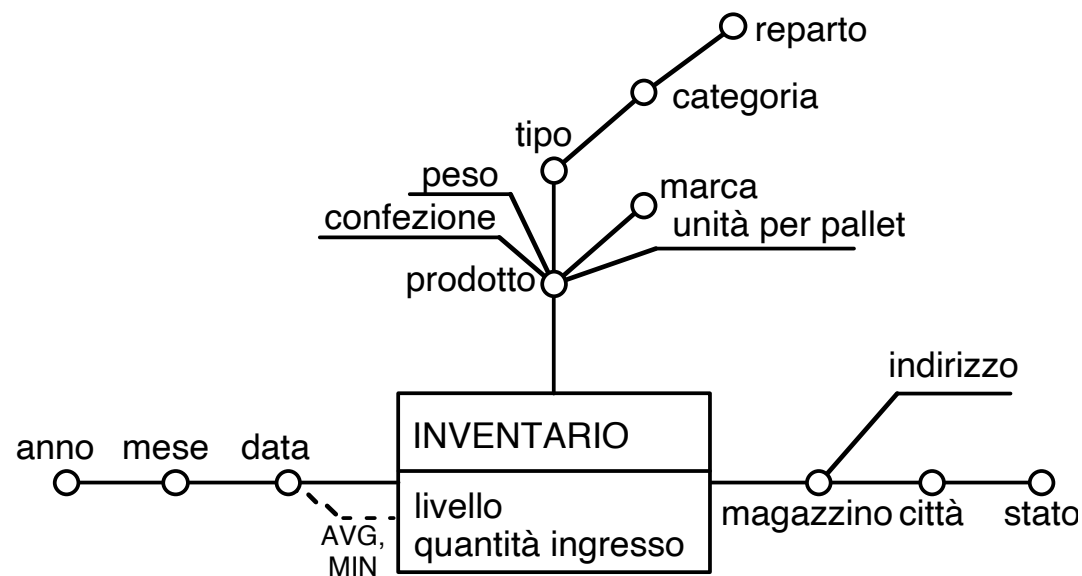
SUM, AVG, MIN, MAX

SUM, AVG, MIN, MAX

AVG, MIN, MAX

Additività

- Una misura è detta **additiva** su una dimensione se i suoi valori possono essere aggregati lungo la corrispondente gerarchia tramite l'operatore di somma, altrimenti è detta **non-additiva**. Una misura non-additiva è **non-aggregabile** se nessun operatore di aggregazione può essere usato su di essa



Misure additive

categoria	tipo	prodotto	1999				2000			
			I '99	II '99	III '99	IV '99	I '00	II '00	III '00	IV '00
pulizia casa	detersivo	Brillo	100	90	95	90	80	70	90	85
		Sbianco	20	30	20	10	25	30	35	20
		Lucido	60	50	60	45	40	40	50	40
	sapone	Manipulite	15	20	25	30	15	15	20	10
		Scent	30	35	20	25	30	30	20	15
alimentari	latticino	Latte F Slurp	90	90	85	75	60	80	85	60
		Latte U Slurp	60	80	85	60	70	70	75	65
		Yogurt Slurp	20	30	40	35	30	35	35	20
	bibita	Bevimi	20	10	25	30	35	30	20	10
		Colissima	50	60	45	40	50	60	45	40

categoria	1999				2000			
	I '99	II '99	III '99	IV '99	I '00	II '00	III '00	IV '00
pulizia casa	225	225	220	200	190	185	215	170
alimentari	240	270	280	240	245	275	260	195

categoria	anno	
	1999	2000
pulizia casa	870	760
alimentari	1030	975

categoria	tipo	anno	
		1999	2000
pulizia casa	detersivo	670	605
	sapone	200	155
alimentari	latticino	750	685
	bibita	280	290

Misure non-additive

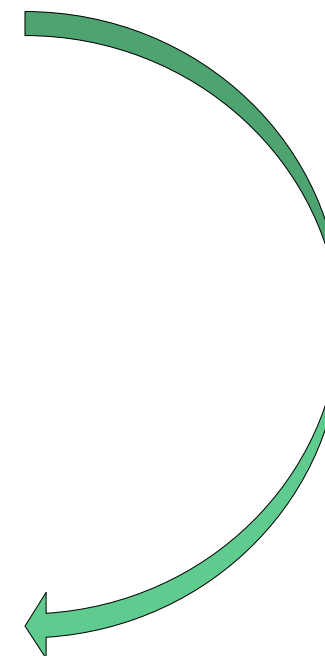
categoria	tipo	prodotto	1999			
			I'99	II'99	III'99	IV'99
pulizia casa	detersivo	Brillo	2	2	2,2	2,5
		Sbianco	1,5	1,5	2	2,5
		Lucido	–	3	3	3
	sapone	Manipulite	1	1,2	1,5	1,5
		Scent	1,5	1,5	2	–



categoria	tipo	1999			
		I'99	II'99	III'99	IV'99
pulizia casa	detersivo	1,75	2,17	2,40	2,67
	sapone	1,25	1,35	1,75	1,50
	<i>media</i>	1,50	1,76	2,08	2,09

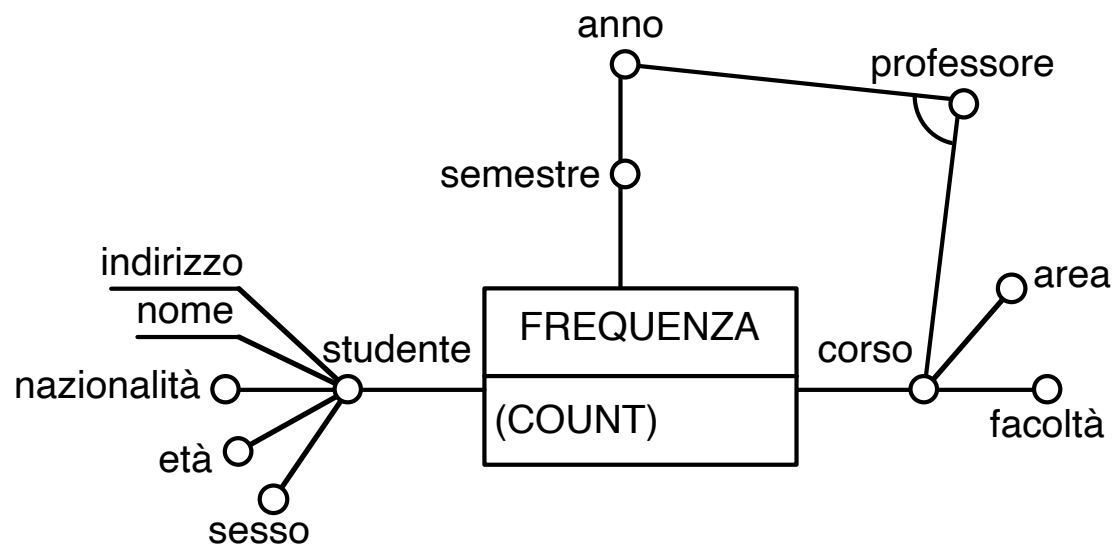


categoria	1999			
	I'99	II'99	III'99	IV'99
pulizia casa	1,50	1,84	2,14	2,38



Schemi di fatto vuoti

- Uno schema di fatto si dice **vuoto** se non ha misure
 - ✓ In questo caso, il fatto registra solo il verificarsi di un evento





Progettazione concettuale: approcci

■ Basata sui requisiti

- ✓ Il progettista deve essere in grado di enucleare, dalle interviste condotte presso l'utente, un'indicazione precisa circa i fatti da rappresentare, le misure che li descrivono e le gerarchie attraverso cui aggregarli utilmente. Il problema del collegamento tra lo schema concettuale così determinato e le sorgenti operazionali viene affrontato in un secondo tempo

■ Basata sulle sorgenti



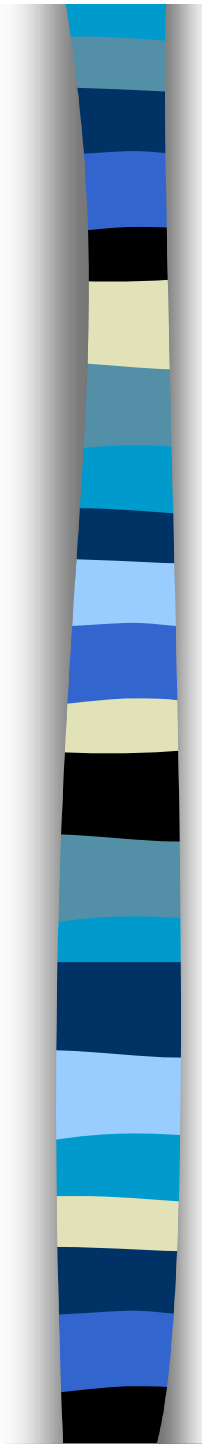
- ✓ È possibile definire lo schema concettuale in funzione della struttura delle sorgenti, evitando il complesso compito di stabilire il legame con esse a posteriori. Inoltre, è possibile derivare uno schema concettuale prototipale dagli schemi operazionali in modo pressoché automatico



Progettazione concettuale: come

- La progettazione concettuale viene effettuata a partire dalla documentazione relativa al database riconciliato:
 - ✓ Schemi E/R
 - ✓ Schemi Relazionali
 - ✓ Schemi XML
 - ✓

- Passi di progettazione:
 - ① Definizione dei fatti
 - ② Per ogni fatto:
 1. Costruzione di un *albero degli attributi*
 2. Editing dell'albero degli attributi
 3. Definizione delle dimensioni
 4. Definizione delle misure
 5. Creazione dello schema di fatto





L'esempio delle vendite (da schema logico)

PRODOTTI (prodotto, peso, dimensione, dieta,
di Marca: MARCHE, di Tipo: TIPI)

NEGOZI (negozio, indirizzo, telefono, respVendite,
(numDistr, stato): DISTRETTI, in Città: CITTÀ)

SCONTRINI (numScontrino, data, negozio: NEGOZI)

VENDITE (prodotto: PRODOTTI, numScontrino: SCONTRINI,
quantità, prezzoUnitario)

MAGAZZINI (magazzino, indirizzo)

CITTÀ (città, regione: REGIONI)

REGIONI (regione, stato: STATI)

STATI (stato)

DISTRETTI (numDistr, stato: STATI)

PROD_IN_MAGAZZ (prodotto: PRODOTTI, magazzino: MAGAZZINI)

MARCHE (codMarca, prodottaIn: CITTÀ)

TIPI (tipo, gruppoMarketing: GRUPPIMARK,
categoria: CATEGORIE)

GRUPPIMARK (gruppoMarketing, responsabile)

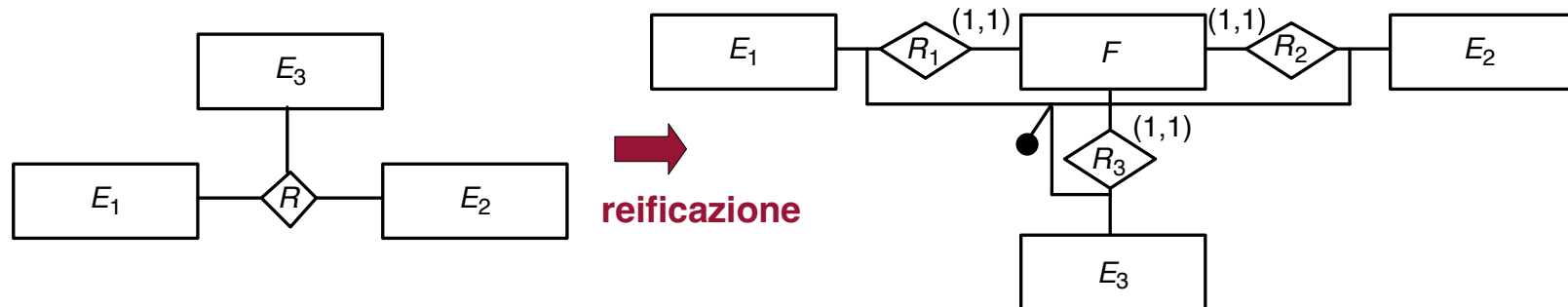
CATEGORIE (categoria, reparto: REPARTI)

REPARTI (reparto, capoReparto)

Definizione dei fatti

I fatti sono concetti di interesse primario per il processo decisionale; tipicamente, corrispondono a eventi che accadono dinamicamente nel mondo aziendale

- Sullo schema E/R un fatto può corrispondere o a un'entità F o a un'associazione n-aria R tra le entità E_1, E_2, \dots, E_n

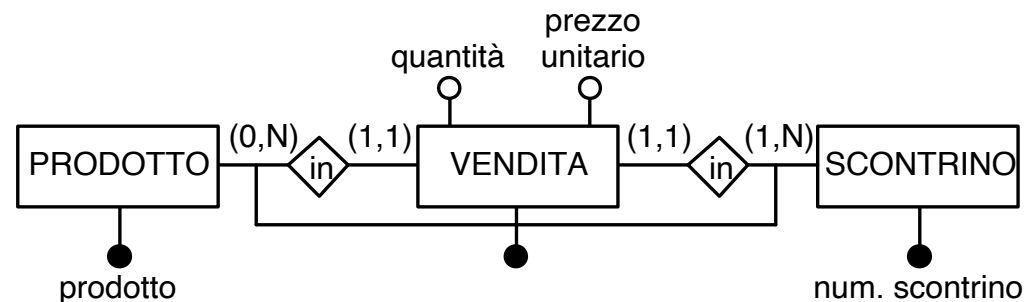


- Sullo schema relazionale un fatto corrisponde a una relazione F

Definizione dei fatti

Le entità o relazioni che rappresentano archivi frequentemente modificati (come VENDITA) sono buoni candidati per definire fatti; quelli che rappresentano archivi quasi-statici (come NEGOZIO e CITTÀ) no

- ✓ Nell'esempio delle vendite si sceglie come fatto l'associazione VENDITA, corrispondente alla relazione VENDITE.



- Ogni fatto identificato diviene la radice di un nuovo schema



Costruzione dell'albero degli attributi

- L'albero degli attributi è un albero in cui:
 - ✓ ogni vertice corrisponde a un attributo - semplice o composto - dello schema sorgente;
 - ✓ la radice corrisponde all'identificatore (chiave primaria) di F ;
 - ✓ per ogni vertice v , l'attributo corrispondente determina funzionalmente tutti gli attributi corrispondenti ai discendenti di v
- L'albero degli attributi corrispondente a F può essere costruito in modo automatico applicando una procedura che naviga ricorsivamente le dipendenze funzionali espresse, nello schema sorgente, dagli identificatori e dalle associazioni a-uno

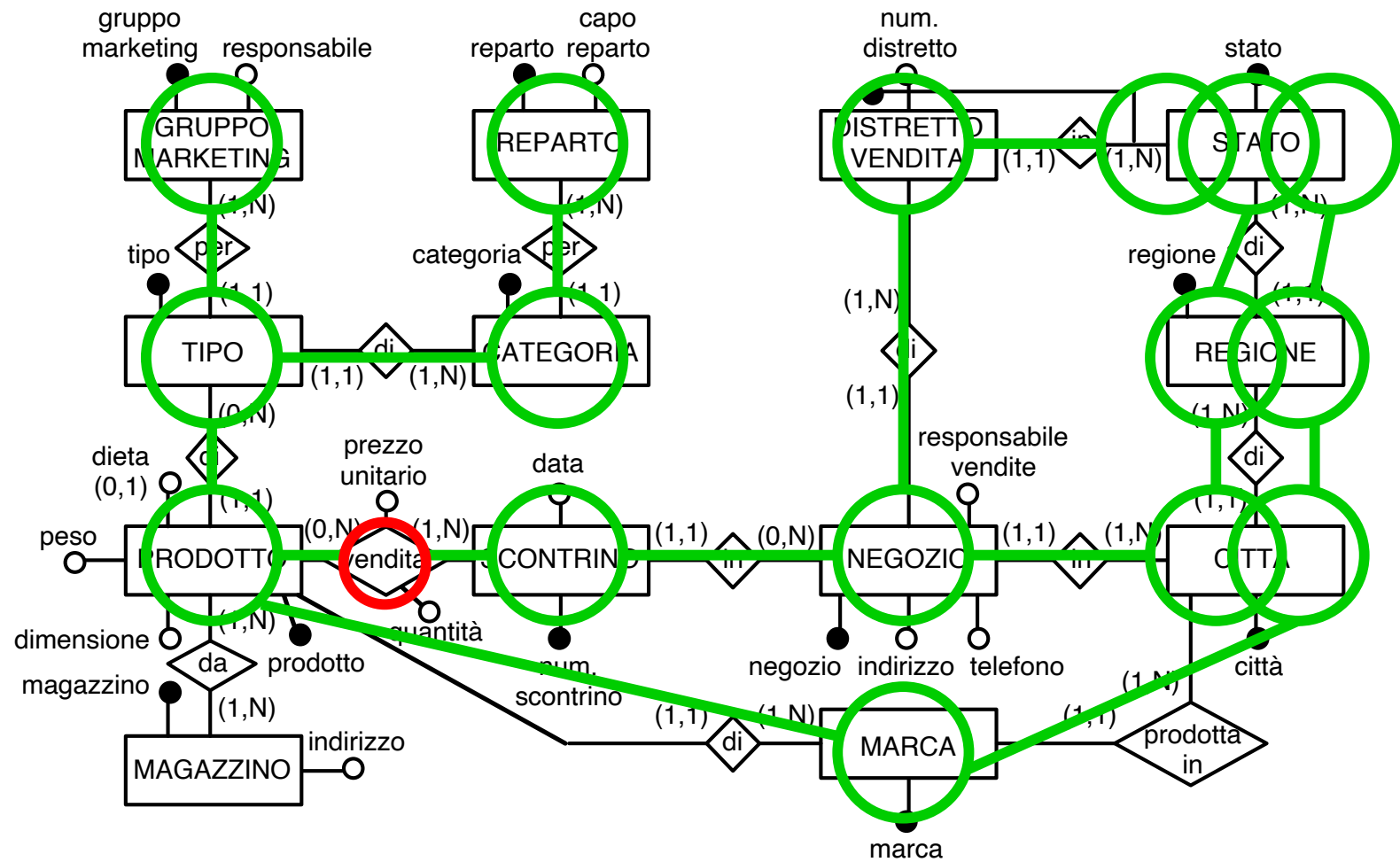
Costruzione dell'albero degli attributi

```
root=nuovoVertice(ident(F));           // ident(F) è l'identificatore di F
// la radice dell'albero è etichettata con l'identificatore dell'entità scelta come fatto
traduci(F, root);
```

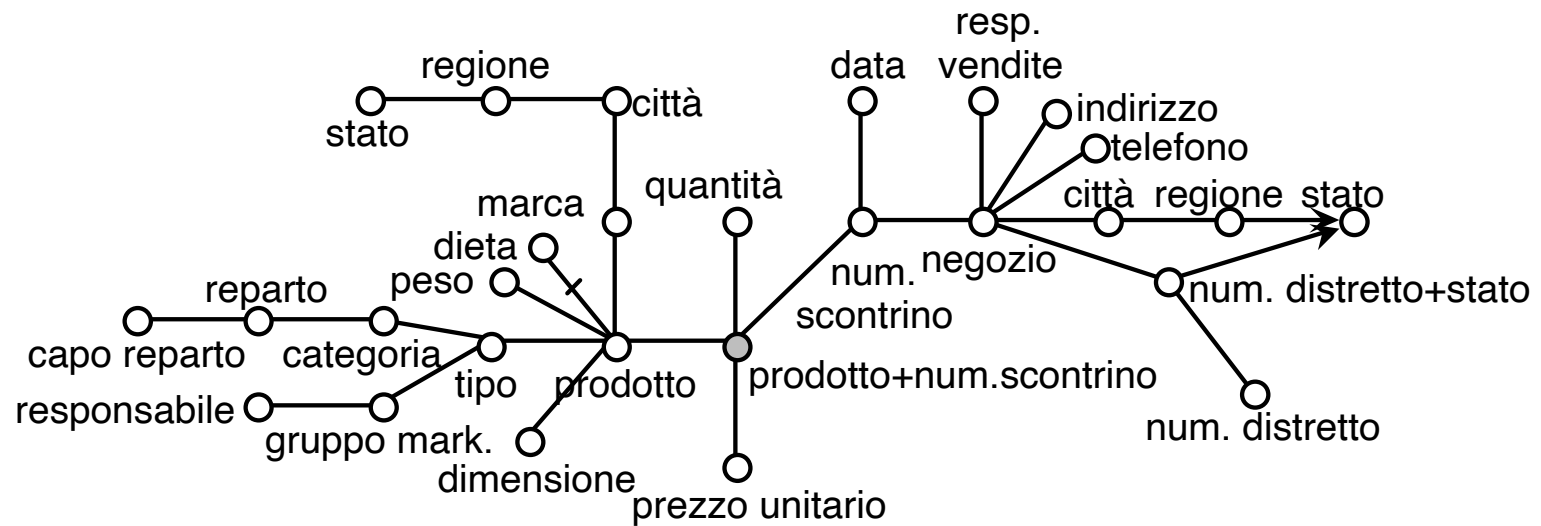
procedura traduci(E, v):

```
// E è l'entità corrente dello schema sorgente, v il vertice corrente dell'albero
{   per ogni attributo  $a \in E$  tale che  $a \neq \text{ident}(E)$ 
    aggiungiFiglio(v, nuovoVertice(a));
    // aggiunge al vertice v un figlio a
per ogni entità G connessa a E da un'associazione R tale che  $\max(E, R)=1$ 
    {   per ogni attributo  $b \in R$ 
        aggiungiFiglio(v, nuovoVertice(b));
        // aggiunge al vertice v un figlio b
        prossimo=nuovoVertice(ident(G));
        // crea un nuovo vertice con il nome dell'identificatore di G ...
        aggiungiFiglio(v, prossimo);
        // ... lo aggiunge a v come figlio ...
        traduci(G, prossimo);
        // ... e innesca la ricorsione
    }
}
```

L'esempio delle vendite



L'esempio delle vendite

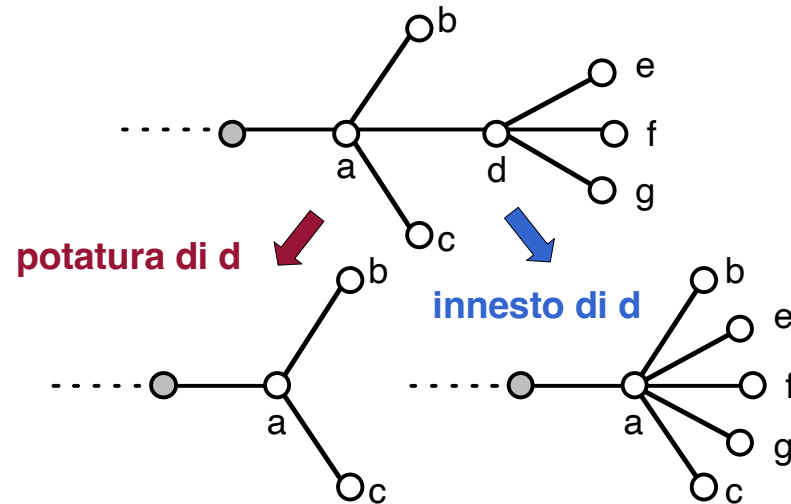




Editing dell'albero

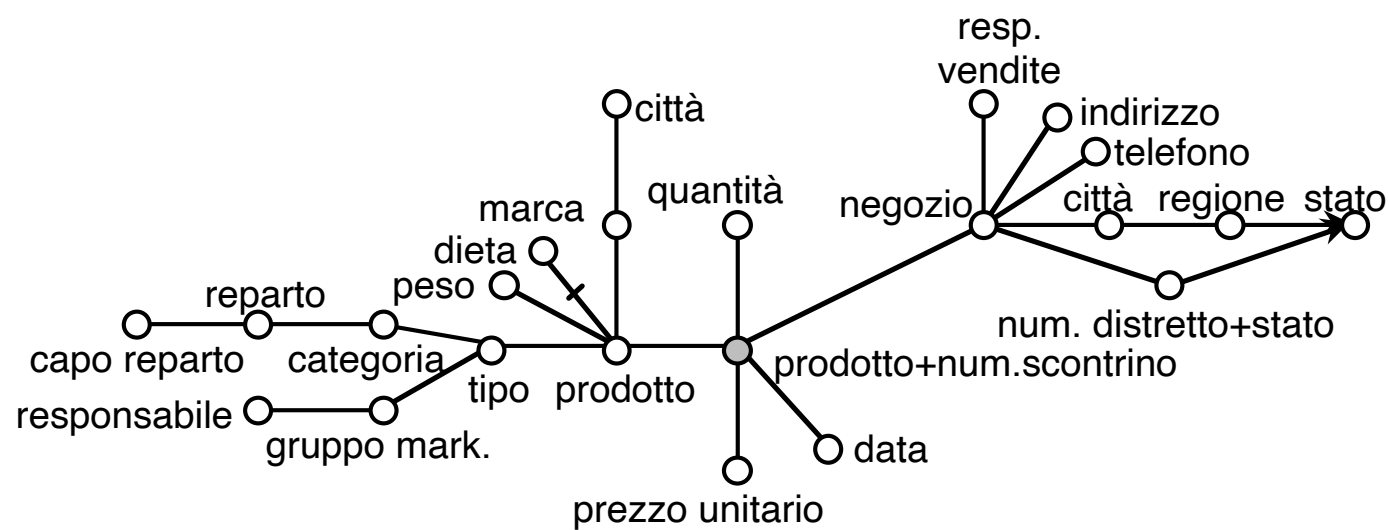
- In genere non tutti gli attributi dell'albero sono d'interesse per il data mart; quindi, l'albero può essere manipolato per eliminare i livelli di dettaglio non necessari
 - ✓ La **potatura** di un vertice v si effettua eliminando l'intero sottoalbero con radice in v
 - Gli attributi eliminati non verranno inclusi nello schema di fatto, quindi non potranno essere usati per aggregare i dati
 - ✓ L'**innesto** viene utilizzato quando, sebbene un vertice esprima un'informazione non interessante, è necessario mantenere nell'albero i suoi discendenti
 - L'innesto del vertice v , con padre v' , viene effettuato collegando tutti i figli di v direttamente a v' ed eliminando v ; come risultato verrà perduto il livello di aggregazione corrispondente all'attributo v ma non i livelli corrispondenti ai suoi discendenti

Editing dell'albero



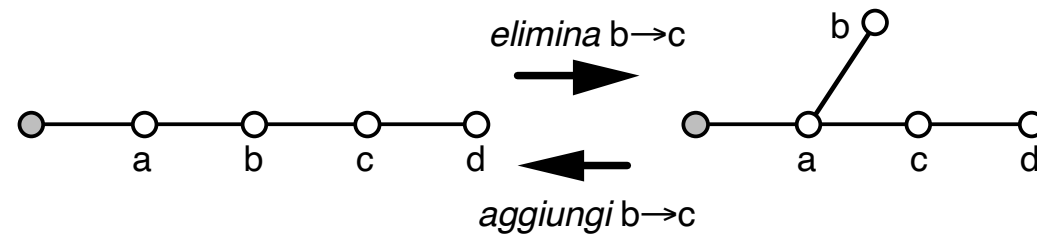
- Quando un vertice opzionale viene innestato, tutti i suoi figli ereditano il trattino di opzionalità
 - ✓ Nel caso di potatura o innesto di un vertice opzionale v con padre v' è possibile aggiungere a v' un nuovo figlio b corrispondente a un attributo booleano che esprima l'opzionalità
- Potare o innestare un figlio della radice che corrisponde, sullo schema sorgente, a un attributo incluso nell'identificatore dell'entità scelta come fatto significa rendere più grossolana la granularità del fatto
 - ✓ Se il vertice innestato ha più di un figlio, si può avere un aumento del numero di dimensioni nello schema di fatto

L'esempio delle vendite

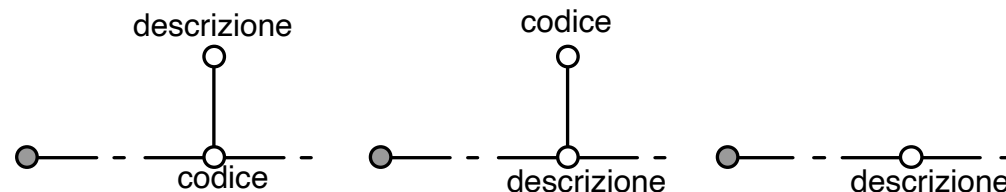


Editing dell'albero

- Nella pratica possono rendersi necessarie ulteriori manipolazioni sull'albero degli attributi
 - ✓ Può essere necessario modificarne radicalmente la struttura sostituendo il padre di un certo nodo: ciò corrisponde ad aggiungere o eliminare una dipendenza funzionale



- ✓ In presenza di un'associazione uno-a-uno sono consigliabili due soluzioni:
 - quando il vertice v determinato dall'associazione uno-a-uno ha dei discendenti di interesse lo si può eliminare dall'albero tramite innesto;
 - quando v non ha discendenti di interesse lo si può rappresentare come attributo descrittivo.
 - in alcuni casi può convenire *invertire* i due nodi coinvolti

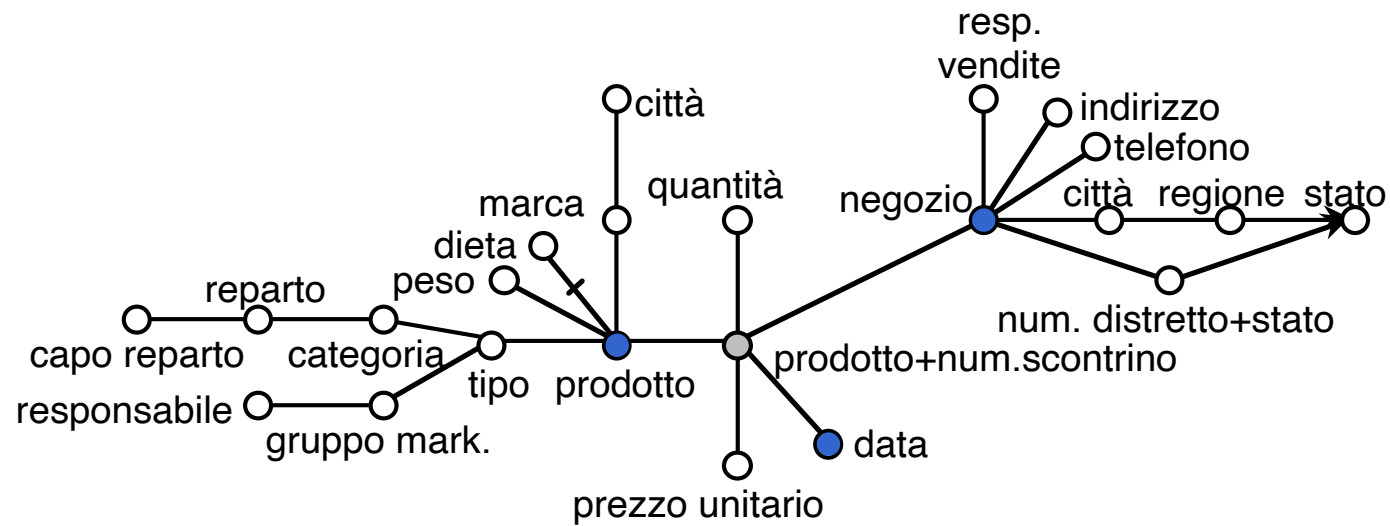




Definizione delle dimensioni

- Le dimensioni devono essere scelte nell'albero degli attributi tra i vertici figli della radice; possono corrispondere ad attributi discreti o a intervalli di valori di attributi discreti o continui
- La loro scelta è cruciale per il progetto poiché definisce la *granularità* degli eventi primari
- Il tempo dovrebbe sempre essere una dimensione:
 - ✓ Se la sorgente è uno schema storico, il tempo è rappresentato esplicitamente come un attributo; se appare nell'albero degli attributi come figlio di un vertice diverso dalla radice, si può effettuare un innesto o eliminare una dipendenza funzionale al fine di farlo diventare un figlio diretto della radice e quindi una dimensione
 - ✓ Nelle sorgenti snapshot il tempo non viene rappresentato esplicitamente; in questo caso il tempo viene tipicamente aggiunto “manualmente” allo schema di fatto

L'esempio delle vendite

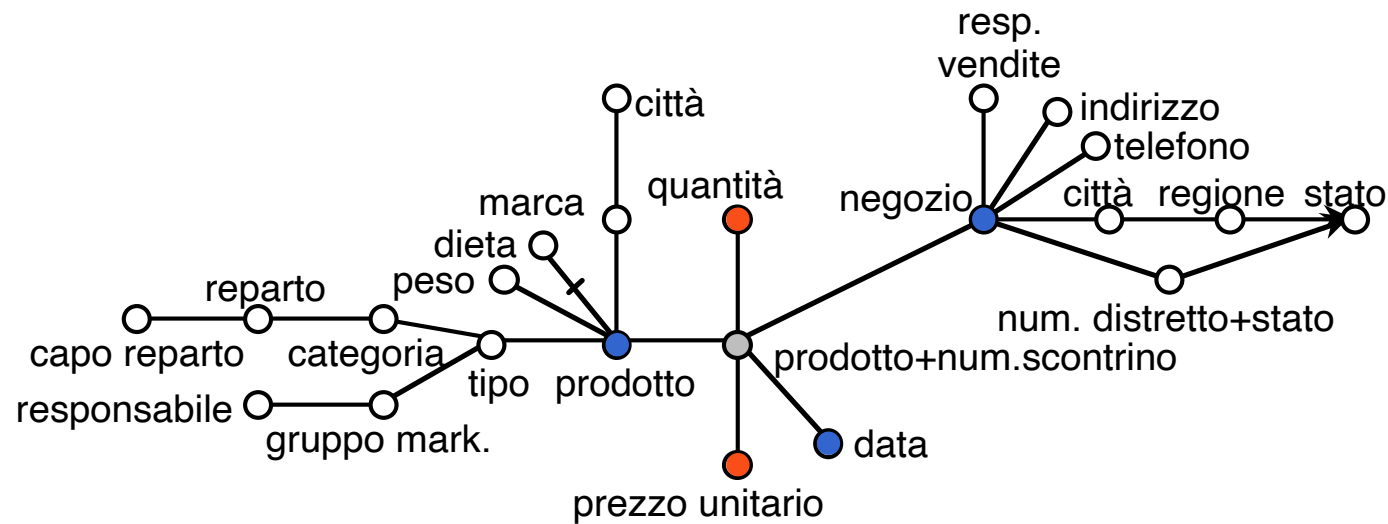




Definizione delle misure

- Se tra le dimensioni compaiono tutti gli attributi che costituiscono un identificatore dell'entità fatto, allora le misure corrispondono ad attributi numerici che siano figli della radice dell'albero
- In caso contrario le misure devono essere definite applicando, ad attributi numerici dell'albero, funzioni di aggregazione che operano su tutte le istanze di F corrispondenti a ciascun evento primario (in genere si tratta di somma/media/massimo/minimo di espressioni oppure del conteggio del numero di istanze di F)
 - ✓ Un fatto può anche non avere misure
 - ✓ Qualora la granularità del fatto sia differente da quella dello schema sorgente, può essere utile definire più misure che aggregano lo stesso attributo tramite operatori diversi

L'esempio delle vendite



GLOSSARIO

quantità venduta = SUM(VENDITA.quantità)

incasso = SUM(VENDITA.quantità*VENDITA.prezzoUnitario)

prezzo unitario = AVG(VENDITA.prezzoUnitario)

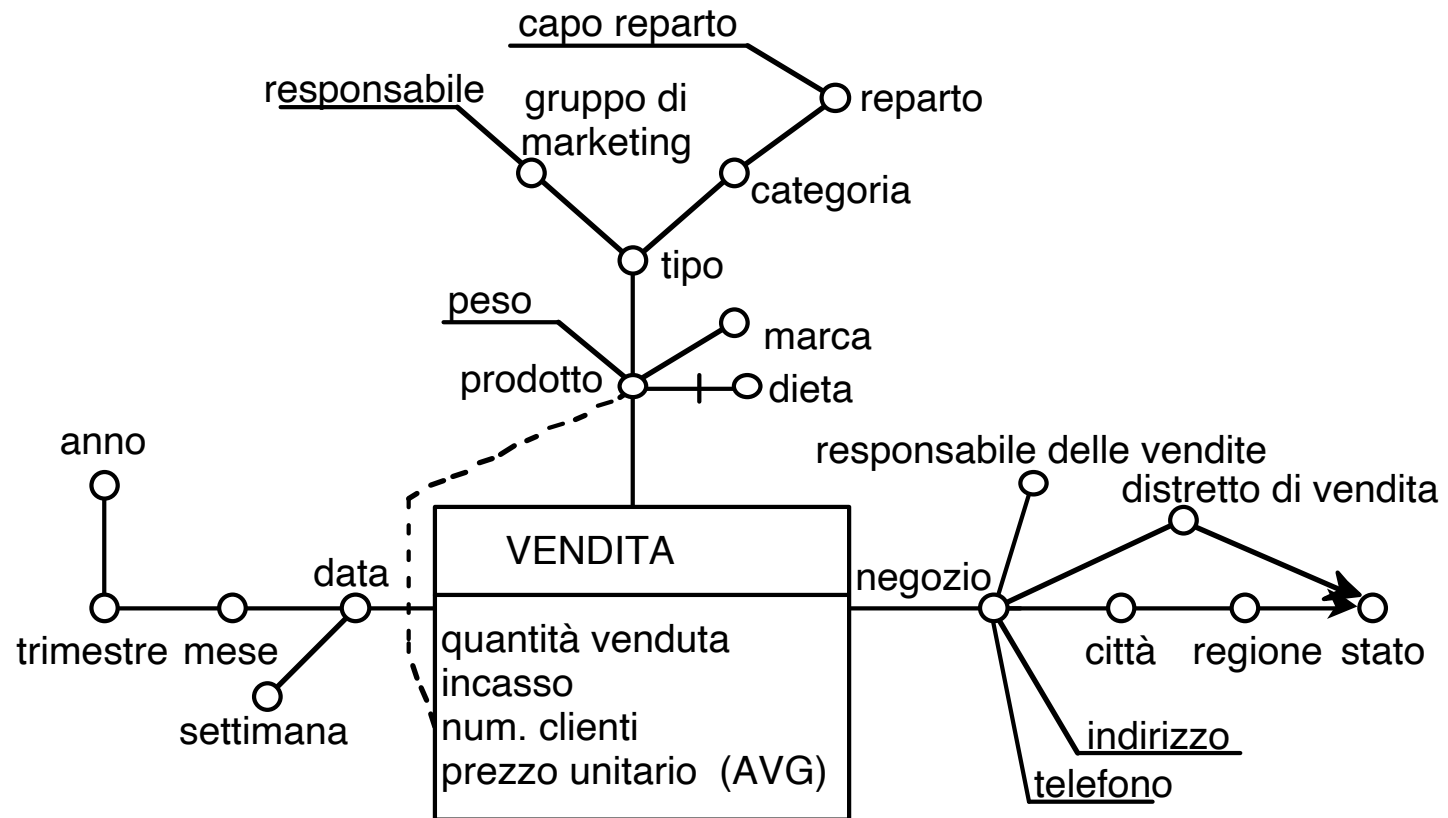
num. clienti = COUNT(*)



Creazione dello schema di fatto

- L'albero degli attributi può ora essere tradotto in uno schema di fatto che include le dimensioni e misure definite
 - ✓ le gerarchie corrispondono ai sottoalberi dell'albero degli attributi con radice nelle diverse dimensioni
 - ✓ il nome del fatto corrisponde al nome dell'entità scelta come fatto
 - ✓ È possibile potare e innestare l'albero per eliminare dettagli inutili
 - ✓ È possibile aggiungere attributi dimensionali definendo opportuni intervalli per attributi numerici (per es. sulla dimensione tempo)
 - ✓ Gli attributi che non verranno usati per l'aggregazione possono essere contrassegnati come descrittivi; tra questi compariranno in genere anche gli attributi determinati da associazioni uno-a-uno e privi di discendenti
 - ✓ Per quanto riguarda eventuali attributi alfanumerici figli della radice ma non prescelti né come dimensioni né come misure:
 - se la granularità degli eventi primari coincide con quella dell'entità F, essi possono essere rappresentati come attributi descrittivi associati direttamente al fatto, di cui descriveranno ciascuna occorrenza
 - se invece le due granularità sono differenti, essi devono necessariamente essere potati

L'esempio delle vendite

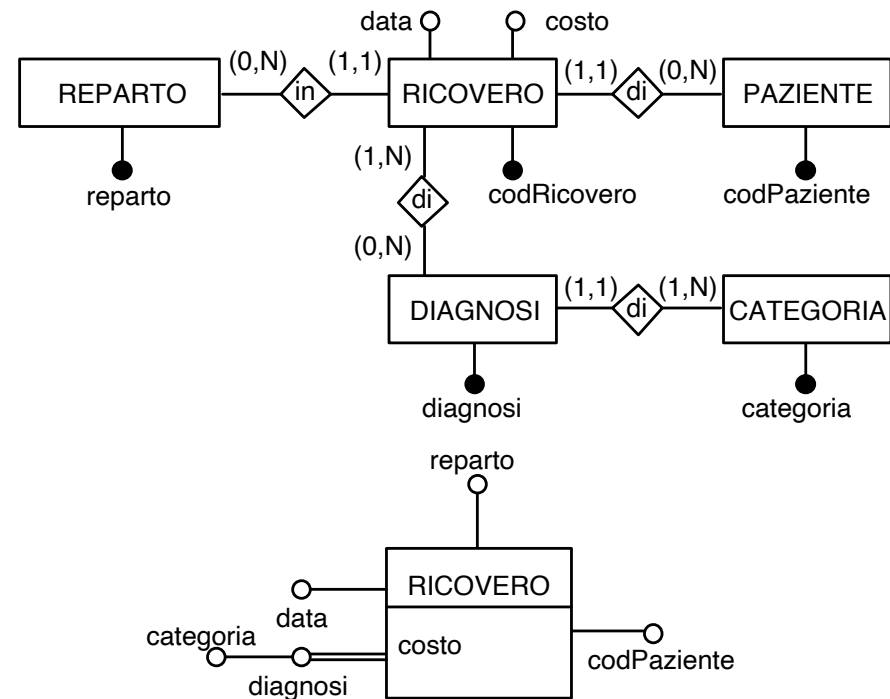
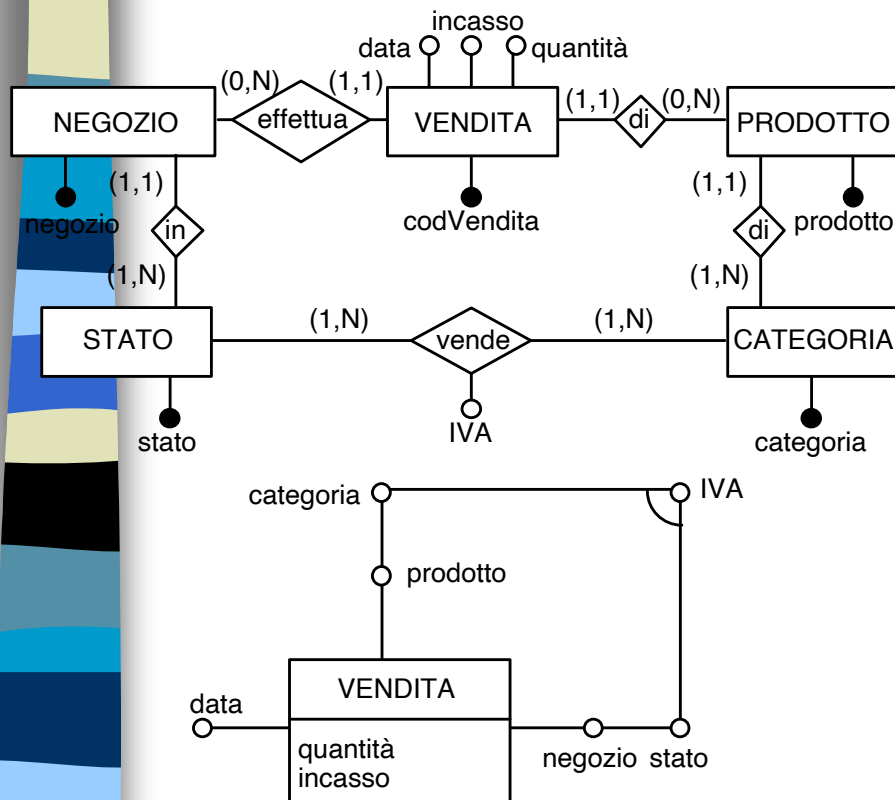




Creazione dello schema di fatto

- Eventuali attributi cross-dimensionali e archi multipli possono essere evidenziati in questa fase
 - ✓ Identificare queste tipologie di attributi a partire dallo schema sorgente è complesso, poiché richiede di navigare anche le associazioni a-molti, per cui si preferisce definirli a partire dai requisiti utente per rappresentarli solo successivamente sullo schema di fatto
 - Un attributo cross-dimensionale corrisponde in genere a un attributo posto su un'associazione molti-a-molti R dello schema E/R; i suoi padri nello schema di fatto corrisponderanno allora agli identificatori delle entità coinvolte in R
 - Un arco multiplo corrisponde a un'associazione a-molti R da un'entità E a un'entità G; nello schema di fatto, esso potrà allora connettere l'identificatore di E o il fatto con un attributo di R o di G

Creazione dello schema di fatto





Creazione dello schema di fatto

- In questa fase devono anche essere identificate le eventuali non-additività e non-aggregabilità presenti nello schema, considerando tutte le accoppiate dimensione-misura
- Dato uno schema di fatto n-dimensionale, per la dimensione d_i e la misura m_j , la domanda da porsi sarà:

“Siano $\{val_1, \dots, val_k\}$ i valori assunti dalla misura m_j nei k eventi primari corrispondenti a k differenti valori presi dal dominio della dimensione d_i e da un valore prefissato di ciascuna delle altre $n-1$ dimensioni. Volendo caratterizzare complessivamente i k eventi con un unico valore di m_j , quali operatori di aggregazione ha senso utilizzare?”