

Modellazione

Schemi e Istanze

schema: sostanzialmente invariante nel tempo, descrive la struttura (aspetto intensionale)

- nel modello relazionale, le intestazioni delle tabelle

istanza: i valori attuali, che cambiano anche molto rapidamente (aspetto estensionale)

- nel modello relazionale, il “corpo” di ciascuna tabella

Modelli

modelli concettuali: permettono di rappresentare i dati in modo indipendente dal sistema

- cercano di descrivere i concetti del mondo reale
- sono utilizzati nelle fasi preliminari di progettazione

modelli logici: utilizzati per l'organizzazione dei dati da parte dei DBMS

- utilizzati dalle applicazioni
- indipendenti dalle strutture fisiche

Motivi per l'uso dei modelli concettuali

Se partiamo a progettare direttamente dallo schema logico della base di dati di un'applicazione:

- da dove cominciamo?
- rischiamo di perderci subito nei dettagli
- dobbiamo pensare subito a come correlare le varie tabelle (chiavi etc.)
- ci sono grosse rigidità

Il modello concettuale ha:

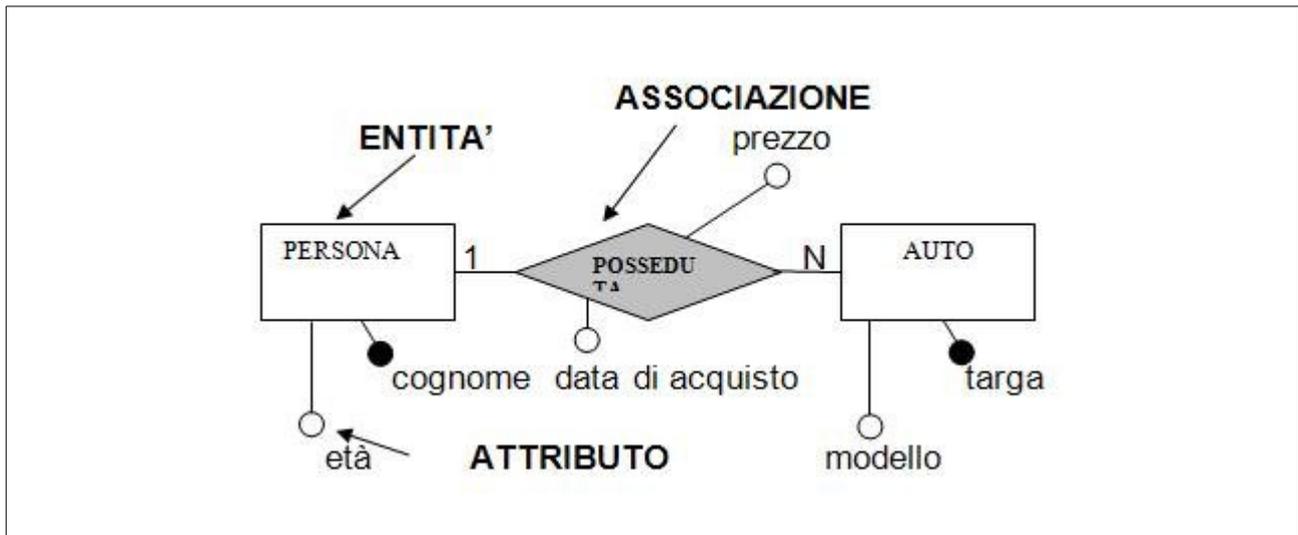
- una rappresentazione grafica (più comunicativa)
- maggiore astrazione rispetto ai dettagli implementativi
- quindi più semplice
- quindi più comprensibile per l'utente/committente del sistema

La tecnica di modellazione concettuale più diffusa è quella “**Entity-Relationship**” (E-R)

Modello Entity-Relationship

Elementi e costrutti del modello E-R:

- **Entità** - rappresentazione di una categoria di oggetti del mondo reale che viene modellato
- **Attributo** – caratteristica dell'entità d'interesse nel modello (aggregabili in attributi composti)
- **Relazioni o associazioni** (Relationship) – collegamento tra diverse entità del modello
- **Cardinalità** – delle relazioni ma anche degli attributi
- **Identificatore** – insieme di attributi e/o relazioni che identificano un'entità
- **Generalizzazione** – classificazione gerarchica delle entità con ereditarietà delle caratteristiche (attributi, relazioni, etc..)



Entità

Di solito si tratta di classi di oggetti (non singoli oggetti ma insiemi di oggetti con caratteristiche comuni) che possiedono un significato ed un'esistenza autonoma

Tutto dipende dal contesto applicativo, quella che può essere un'entità per un progetto potrebbe non esserlo in un altro (persone come entità in un sistema anagrafico, come valore numerico in un sistema geografico).

Ogni singola entità è indicata con un nome, espressivo all'intero del contesto applicativo.

Si seguono di solito delle opportune convenzioni, come quella di utilizzare nomi al singolare.

Attributi

Sono in prima approssimazione i campi di una tabella.

Possono anche essere **COMPOSTI**.

Relationship

Legame logico fra due o più entità, **rilevante** nell'applicazione di interesse

Il legame padre-figlio è fondamentale in un sistema per l'anagrafe civile, ininfluente in un sistema di gestione clienti.

Anche le relazioni hanno un nome, che sia possibilmente significativo e possibilmente non un verbo ("succedere", "contiene", ...) ma un sostantivo ("successione", "componente", ...).

E' possibile definire per le varie entità che sono collegate da una relazione dei ruoli, che appunto dettagliano qualora sia necessario che parte gioca la singola entità nella relazione.

Tra due entità possono esservi più relazioni, ognuna con un suo particolare significato (ad esempio tra le entità "persona" e "città" possono esserci le relazioni "Residenza", "Nascita", "Domicilio",...).

Le relazioni possono essere tra più di due entità, ed in alcuni casi possono essere "opzionali" tra due o più entità.

Le relazioni possono essere anche ricorsive sulla stessa entità.

In questo caso di solito è necessario definire i ruoli all'interno della relazione che altrimenti potrebbe essere interpretata nel verso sbagliato (ad esempio per la relazione "genitore" sull'entità "persona" dev'essere indicato quale è la "madre" e quale il "figlio").

Promozione di una relazione

Un attento esame delle relazioni a volte porta a “promuovere” una relazione e trasformarla in un'entità.

Ad esempio un esame visto in un primo momento come una relazione tra studente e corso può poi diventare per esigenze applicative un'entità autonoma con una relazione con studente ed una con corso.

Di solito questo avviene quando ad una relazione iniziano ad essere associati molti attributi.

Cardinalità

Indica genericamente la numerosità di un ente.

Di due tipi:

- cardinalità di una relazione
- cardinalità di un attributo

La **cardinalità di una relazione** è una coppia di numeri interi che indica, per ogni “ramo” di una relazione (cioè per ogni entità che appare nella relazione), il numero minimo e massimo di volte che una relazione può esistere per una particolare occorrenza (oggetto) di un'entità.

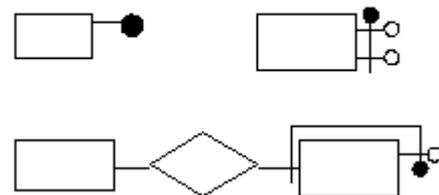
Ad esempio la relazione ricorsiva “genitore” avrà una cardinalità 0-n per il ramo “madre” (cioè una persona può essere “madre” da 0 a n volte), mentre avrà una cardinalità 1-1 per il ramo “figlio” (cioè una persona deve sempre essere “figli” una ed una sola volta).

Identificatore

Insieme di attributi e/o di relazioni che identifica un'entità.

Gli identificatori sono i candidati a diventare le chiavi.

Identificatore Interno ed Esterno



Generalizzazione

Una serie di entità potrebbero avere delle caratteristiche in comune, come ad esempio un impiegato, uno studente, un professore: hanno tutte un nome, un cognome, una data di nascita, etc.

In questo caso si può pensare ad un'entità "Persona" che è la generalizzazione di tutte queste entità e definire che le varie impiegato, studente e professore derivano da questa entità generale, ereditando da essa tutte le caratteristiche e definendone eventualmente di nuove.

Definitions:

entity something about which data is collected, stored, and maintained

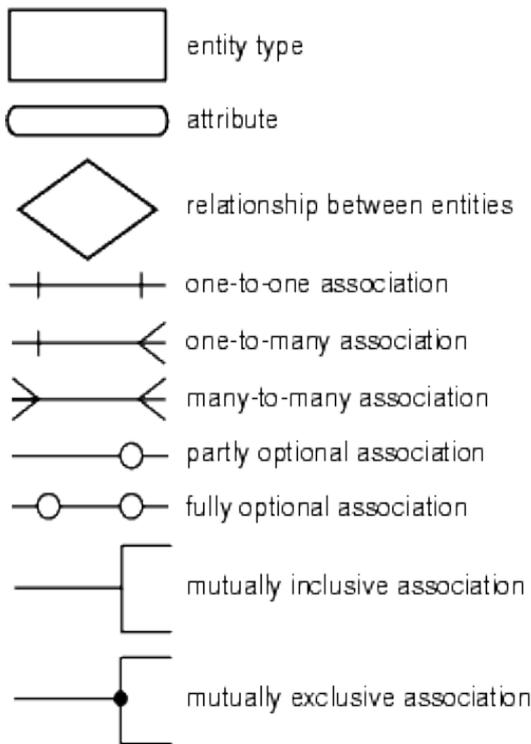
attribute a characteristic of an entity

relationship an association between entities

entity type a class of entities that have the same set of attributes

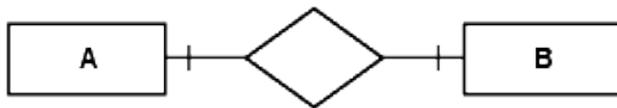
record an ordered set of attribute values that describe an instance of an entity type

Symbols:

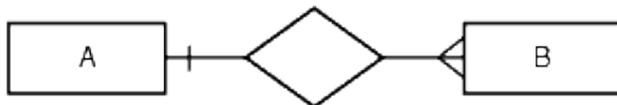


Examples:

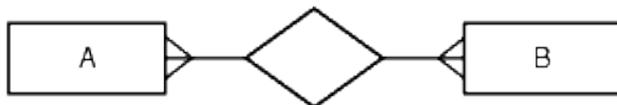
One A is associated with one B:



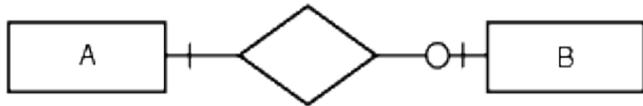
One A is associated with one or more B's:



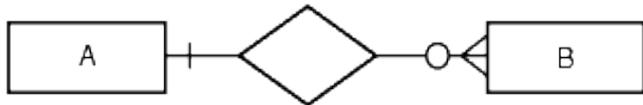
One or more A's are associated with one or more B's:



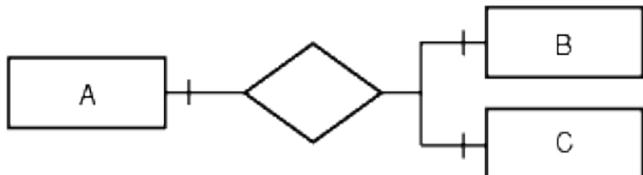
One A is associated with zero or one B:



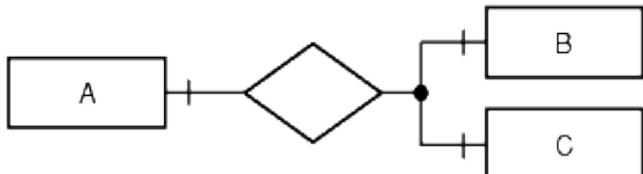
One A is associated with zero or more B's:



One A is associated with one B and one C:

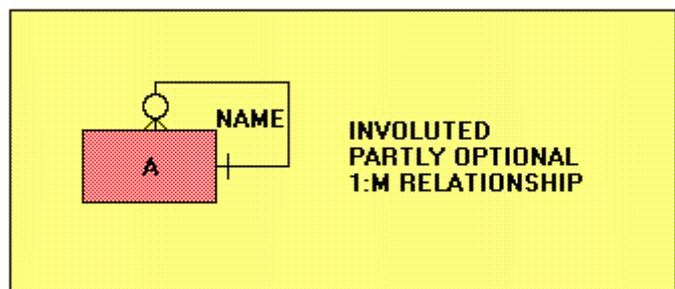
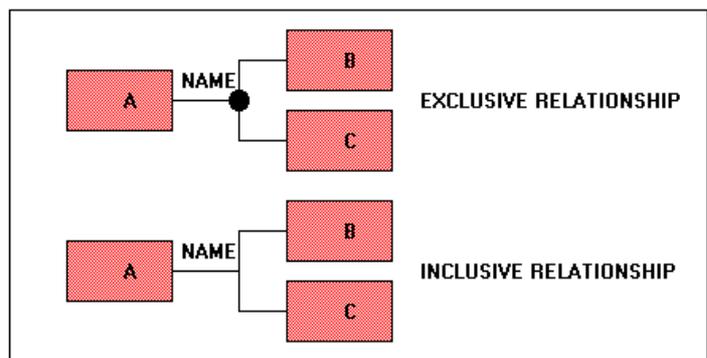


One A is associated with one B or one C (but not both):



<p>DEPARTMENT</p> <p>DepartmentName</p> <p>BudgetCode</p> <p>OfficeNumber</p>	<p>DEPARTMENT entity; DepartmentName is identifier; BudgetCode and OfficeNumber are attributes.</p>
	<p>1:1, nonidentifying relationship. A relates to zero or one B; B relates to exactly one A. Identifier and attributes not shown.</p>
	<p>1:N, nonidentifying relationship. A relates to one or many Bs; B relates to zero or one A. Identifier and attributes not shown.</p>
	<p>Many-to-many, nonidentifying relationship. A relates to zero or more Bs; B relates to one or more As.</p>
	<p>1:N identifying relationship. A relates to zero, one, or many Bs. B relates to exactly one A. Identifier and attributes not shown. For identifying relationships, the child must always relate to exactly one parent. The parent may relate to zero, one, many, or a combination of these minimum cardinalities.</p>
	<p>A is supertype, C and D are exclusive subtypes. Discriminator not shown. Identifier and attributes not shown.</p>
	<p>A is supertype, C and D are inclusive subtypes. Identifier and attributes not shown.</p>

Symbol	Meaning
	One-Mandatory
	Many-Mandatory
	One-Optional
	Many-Optional



METODOLOGIA DI DISEGNO

Concetti di base del processo di disegno di un DB

Passi essenziali di disegno

Il disegno di un DB può essere effettuato seguendo una serie di passi elementari, cioè “eseguendo” una specie di algoritmo.

Come già più volte detto non esiste un algoritmo unico e perfetto per il disegno di un DB a partire da delle specifiche testuali, sia perché esistono diversi metodi e strategia che possono essere vantaggiosi o problematici a seconda del problema specifico che si deve affrontare, ma anche per la diversa forma che può assumere la descrizione testuale di uno stesso problema, con maggiore o minore rigore formale.

Di seguito si fornisce quindi una traccia più che un algoritmo, che dev'essere seguita sempre con un forte spirito critico in modo da non farsi fuorviare del metodo rispetto al contesto specifico del problema.

1. Leggere il testo
2. Individuare le entità
3. Verificare se non si sono entità duplicate
 - Eventuale cambio dei nomi per semplicità
4. Individuare le associazioni tra le entità
5. Verificare se alcune associazioni hanno attributi
6. Calcolare la cardinalità delle associazioni
7. Trovare gli attributi espliciti delle entità
8. Definire le chiavi primarie delle entità
9. Identificare eventuali Gerarchie
10. Stesura dello schema E-R

Leggere il testo

Senza un'attenta lettura ed una comprensione complessiva del testo non si può sperare di creare un disegno corretto e soprattutto corrispondente al contenuto del testo.

Spesso una sola lettura, una singola passata, non è sufficiente per una piena comprensione, ed in alcuni casi può essere necessario rivolgersi a chi ha scritto il testo per comprendere meglio alcuni dettagli che risultano non chiari. Questo può essere dovuto a diversi fattori, che vanno dalla non condivisa base di conoscenza sul problema affrontato dal testo alla difficoltà intrinseca di tradurre in poche e semplici parole concetti che richiederebbero approfondite analisi. Un caso tipico è quello

di concetti che lo scrittore dà per impliciti in quanto facenti parte del dominio delle conoscenze standard di un certo contesto operativo, ma che non sono poi di comune dominio e quindi immediatamente accessibili al lettore.

Individuare le entità

Verificare se non si sono entità duplicate

Eventuale cambio dei nomi per semplicità

Individuare le associazioni tra le entità

Verificare se alcune associazioni hanno attributi

Calcolare la cardinalità delle associazioni

Trovare gli attributi espliciti delle entità

Definire le chiavi primarie delle entità

Identificazione di eventuali gerarchie

Stesura dello schema E-R

Ristrutturazione degli schemi E-R

Serve per:

- facilitare la successiva traduzione dello schema E-R in uno schema relazionale.
- ottimizzazione delle prestazioni

Passi di ristrutturazione

- analisi delle ridondanze
- trasformazione delle generalizzazioni
- eliminazione attributi multi-valore
- accorpamento entità (associazioni 1-1)
- partizionamento verticale di entità (parte dei dati di un'entità non vengono quasi mai usati)
- partizionamento orizzontale di associazioni multiple
- identificatori principali

La ristrutturazione deve essere guidata di due obiettivi:

1. pulizia del disegno
2. ottimizzazione delle prestazioni

Questi due obiettivi spesso vanno di pari passo, ma a volte divergono, ed è quindi necessario trovare il giusto equilibrio tra “perfezione formale” del disegno e “sporca ottimizzazione” delle operazioni.

Analisi delle ridondanze

Le ridondanze creano problemi di allineamento dei dati, ma in alcuni casi possono velocizzare le interrogazioni.

Vantaggi:

- interrogazioni più semplici

Svantaggi:

- complicazione ed appesantimento degli aggiornamenti e **rischio di inconsistenza**
- maggiore occupazione di spazio

Le ridondanze possono essere dovute:

- attributi ripetuti tra entità
- attributi i cui valori possono essere altresì derivati da altri attributi della stessa entità
- attributi i cui valori possono essere derivati da attributi di altre entità
- attributi i cui valori possono essere derivati da conteggi su entità
- associazioni derivabili da altre associazioni

Trasformazione delle generalizzazioni

Le generalizzazioni possono essere trasformate secondo tre linee principali, in base alle tipologie di

partizionamento delle informazioni tra entità generale ed entità di dettaglio, sia dal punto di vista della completezza che dell'uso che ne viene fatto.

Le tre linee guida sono:

1. un'unica tabella che accorpa entità generale e tutte le tipologie di dettaglio (minimo comune multiplo, 1 sola tabella) – necessario metodo per identificare il tipo
2. replica della struttura del generale nelle varie entità di dettaglio (n tabelle)
3. entità separate per parte generale e per i vari dettagli (1 + n tabelle)

Ogni linea presenta vantaggi e svantaggi che vanno pesate sulla base dell'uso che si fa dei dati.

1. conviene se gli accessi al generale ed al dettaglio sono contestuali
2. conviene se gli accessi ai dettagli sono distinti per tipologia
3. conviene se gli accessi ai dettagli sono separati dagli accessi alle informazioni generali

Eliminazione attributi multi-valore

Gli attributi multi-valore possono essere mappati con una tabella di dettaglio con una associazione uno a molti con quella di origine.

Intrinsecamente i campi multi-valore hanno dei limiti dal punto di vista della numerosità massima gestibile e dell'accesso al singolo valore, problemi che possono essere superati trasformando appunto in una tabella di dettaglio che mantiene una dipendenza con l'entità principale.

Accorpamento entità (associazioni 1-1) - Partizionamento verticale di entità

Ogni volta che si presenta una associazione 1-1 tra due entità si deve valutare se tale associazione non nasconda in realtà un'identità tra le due entità coinvolte, cioè che le due entità modellate rappresentino aspetti diversi della stessa entità reale.

Dev'essere quindi valutata attentamente l'opportunità di accorpare in un'unica entità gli attributi delle due entità di partenza, tenendo presente che a volte considerazioni di carattere prestazionale possono spingere a non effettuare tale operazione.

Nel senso opposto le varie entità devono essere analizzate in funzione delle interrogazioni svolte su di esse in modo da ottimizzarne l'accesso.

Infatti se una parte dei dati non viene quasi mai acceduta può essere “parcheggiata” in una entità associata che pur mantenendo le informazioni non appesantisce le interrogazioni “abitudinali”.

Oppure se alcune interrogazioni si focalizzano su una parte di dati ed altre su una parte diversa, queste parti possono essere focalizzate “specializzando” due sotto entità in funzione delle interrogazioni.

Partizionamento orizzontale di associazioni multiple

Una associazione 1-n o n-m può avere caratteristiche di accesso particolari, cioè il legame può essere prioritario su una parte di questa associazione, soprattutto quando tale associazione descrive la storia di un'entità o di una associazione che varia nel tempo. Ad esempio una associazione tra un'automobile e tutti i suoi proprietari avrà una focalizzazione sul proprietario attuale, essendo gli altri temporalmente superati.

In tal caso si può prendere in considerazione l'opportunità di dividere la relazione originaria in due relazioni che rappresentano una la caratteristica prioritaria (ad esempio la situazione corrente) e l'altra tutte le caratteristiche secondarie (ad esempio lo storico di tutte le relazioni).

Identificatori principali

Un passo fondamentale è l'analisi delle entità per l'individuazione degli identificatori principali (o chiavi) delle entità stesse.

I criteri di base per l'individuazione degli attributi adatti sono:

- assenza di opzionalità (cioè not null)
- minimalità (dannoso aggiungere attributi superflui)
- valutazione dell'utilizzo relativamente alle varie operazioni

Se nessun insieme di attributi soddisfa queste caratteristiche si può ricorrere alla creazione di nuovi attributi ad hoc: i **codici identificativi**.

Tali codici assumono valori generati appositamente, di solito attraverso **SEQUENCE**

Traduzione in modello relazionale

Passi di traduzione

- entità
- associazioni molti a molti
- associazioni uno a molti
- associazioni uno a uno
- associazioni multiple
- identificatori esterni di entità

Entità

Le entità si mappano direttamente in relazioni che hanno come attributi gli stessi dell'entità originante.

Associazioni molti a molti

Si trasformano in una relazione che ha come attributi le chiavi delle due entità che sono collegate da tale associazione, oltre agli attributi della associazione stessa.

L'unica limitazione è nella cardinalità della associazione poiché i valori minimi e massimi possono essere imposti solo attraverso complicati costrutti.

Associazioni uno a molti

L'associazione viene espressa come attributo nella relazione che partecipa con cardinalità maggiore, che referenziano le chiavi dell'altra entità, più gli eventuali attributi dell'associazione stessa.

Infatti date le due entità coinvolte A e B con cardinalità 1 ed n, l'associazione esisterà al massimo per ogni occorrenza dell'entità B, cioè per ogni tupla della relazione che mappa B.

Se l'associazione ha cardinalità minima 0, cioè non è obbligatoria, gli attributi di chiave e dell'associazione saranno “nullable”.

Associazioni uno a uno

Tale tipo di associazione può essere:

- fusa in una delle due entità (relazioni) coinvolte
- portare a fondere completamente anche le due entità dell'associazione

La scelta dipende dalle caratteristiche applicative di accesso ai dati.

Associazioni multiple

Le associazioni multiple vengono tradotte in modo simile a quanto fatto per le associazioni più semplici, con l'accortezza di identificare quali sono le relazioni che meglio si adattano ad essere usate come punto di aggregazione per le chiavi delle altre relazioni.

Identificatori esterni di entità

L'identificatore della relazione verrà definito aggiungendo anche l'identificatore della associazione esterna

Progettazione

Considerazioni sui costi e sugli aspetti di progettazione, realizzazione e produzione per sistemi: manifatturieri, informatici, di definizione dei dati.

- *Prodotti manifatturieri: peso preponderante sulla fase di produzione di serie (anche se aumentano sempre più i costi di progettazione)*
- *Prodotti informatici: peso quasi paritetico tra progettazione e produzione del primo singolo manufatto (costo di produzione di serie asintoticamente nullo)*

Modello dati: peso quasi completamente sulla parte progettuale (il processo di produzione è quasi sempre automatizzabile con strumenti CASE e comunque il prodotto è il risultato della progettazione)

Strategie di progetto in generale:

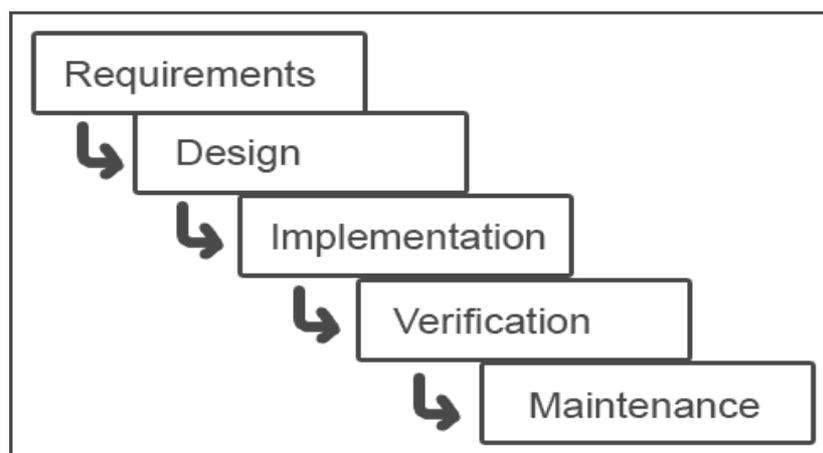
- Top-down
- Bottom-up
- Inside-out
- Prototipale (non su libro)
- Ciclica (non su libro)
- eXtreme Programming (XP - non su libro)

Per la progettazione dei dati non si possono adottare soluzioni prototipali, ne tanto meno XP.

Per i dati la strategia è tipicamente top-down ciclica, cioè fasi di progettazione top-down da un riesame della correttezza e coerenza delle definizioni top sulla base dei risultati down.

Top-down

Modello più anziano, detto anche “a cascata”.



Presuppone la possibilità di completare una fase senza più necessità di riesame sulla base delle attività successive, quindi:

- perfetta conoscenza del dominio
- stabilità dei requisiti
- nessun effetto retroattivo tra dettaglio e generale (il dettaglio non nasconde nulla che non sia

già definito a livello generale)

L'approccio top-down enfatizza la pianificazione ed una completa comprensione del sistema. È ovvio che nessuna codifica può iniziare finché non si è raggiunto almeno un sufficiente livello di dettaglio nella progettazione di una parte significativa del sistema. Questo, comunque, ritarda la fase di test finché una parte significativa della progettazione non è stata completata.

Bottom-up

Dopo una fase di “break-down” in cui il sistema viene spezzettato in elementi sempre più piccoli non ulteriormente scomponibili, si analizzano gli atomi per poi procedere ad aggregazioni successive fino ad ottenere il modello completo del sistema.

Aggregando i sistemi parziali si possono individuare:

- T1 - oggetti che hanno tutte le caratteristiche in comune (unificazione di entità)
- T2 - oggetti di sotto-sistemi diversi che hanno un legame (relazione)
- T3 - oggetti di sotto-sistemi diversi che sono casi particolari di un concetto più generale (generalizzazione)
- T4 e T5 - serie di attributi ricorrenti nei vari sotto-sistemi che possono essere aggregati in un'unica entità o relazione

E' un approccio da chimica-molecolare, che presuppone:

- la possibilità d'individuare subito tutte le componenti del sistema
- che il totale non abbia proprietà che trascendono la somma delle parti (problema della vita: come la si spiega partendo dalla sola chimica molecolare ?)

E' una ottima strategia per affrontare sistemi di grossissime dimensioni in cui un singolo analista non può essere in grado di dominare ogni dettaglio, ma l'integrazione dei modelli è sempre molto complessa ed è soggetta alla capacità di mediazione e di contrattazione tra i vari analisti.

Strategia Bottom-up per i DB

1. Identificare i Tipi d'Entità e, se non esistono, inserire in un Glossario le relative Definizioni
2. Identificare i Tipi di Relazione e, se non esistono, inserire nel Glossario le relative Definizioni
3. Per ogni Tipo d'Entità e ogni Tipo di Relazione, identificare gli Attributi e, se non esistono, inserire nel Glossario le relative Definizioni
4. Introdurre i vincoli di integrità ed eventuali altri vincoli

L'approccio bottom-up enfatizza la codifica e la fase di test precoce, che può iniziare appena il primo modulo è stato specificato. Questo approccio, comunque, induce il rischio che i moduli possano essere codificati senza avere una chiara idea di come dovranno essere connessi ad altre parti del sistema, e quel tipo di link potrebbe non essere facile.