

CAPITOLO 6 : Metodologie e modelli per il progetto

6.1 Introduzione alla progettazione

6.1.1 Il ciclo di vita dei sistemi informativi

Progettazione di basi di dati = solo 1 delle componenti del processo di sviluppo di un sistema informativo => va inquadrata nel ciclo di vita

1. Studio di fattibilità: definisce i costi delle varie alternative e le priorità di realizzazione
2. Raccolta e analisi dei requisiti: che cosa il sistema deve fare
3. Progettazione: dei dati e delle applicazioni
4. Implementazione: realizzare il sistema informativo
5. Validazione e collaudo
6. Funzionamento: si fa solo manutenzione, a meno di modifiche strutturali

A noi interessa di questa catena solamente la parte relativa alle basi di dati, in particolare il 3° passo.

6.1.2 Metodologie di progettazione e basi di dati

Metodologia

1. decomporre in passi;
2. strategie e criteri sui passi;
3. modelli di riferimento per descrivere dati in ingresso ed in uscita

Proprietà di una metodologia

1. generalità
2. qualità: completezza, correttezza, efficienza
3. facilità d'uso

La metodologia che usiamo noi è composta da 3 fasi:

1. Progettazione concettuale
 - IN = requisito
 - OUT = Schema Concettuale (ER)
2. Progettazione logica
 - OUT = Schema Logico (Tabelle)
3. Progettazione Fisica
 - OUT = Schema Fisico

Progettazione concettuale

Si parte da una descrizione informale dei requisiti, e si produce uno **schema concettuale**

, che si riferisce ad un **modello concettuale**

Si rappresenta il **contenuto informativo** della base di dati, e non si pensa a come essa verrà realizzata (ci si pensa dopo)

Progettazione logica

Traduco lo schema ER della progettazione concettuale nel modello di rappresentazione da noi scelto, cioè quello relazionale. Produce lo **schema logico** e fa riferimento ad un **modello logico** dei dati.

Progettazione fisica

Dipende dallo specifico DBMS scelto e si basa su quel sistema.

Specifiche

Ci sono specifiche sui dati e specifiche sulle operazioni.

Nella rappresentazione concettuale si usano soprattutto quelle sui dati, quelle sulle operazioni servono per verificare che lo schema sia completo.

Nella progettazione logica lo schema concettuale in ingresso si somma alle specifiche sulle operazioni per ottenere uno schema logico, conoscendo il modello logico adottato (ma non ancora il DBMS specifico).

Nella progettazione fisica si usa lo schema logico e le specifiche sulle operazioni per ottimizzare le prestazioni del sistema, e ciò dipende dal DBMS usato.

Risultato della progettazione = schema fisico + schema logico + schema concettuale.

6.2 Il modello Entità – Relazione

È un modello concettuale, e fornisce costrutti per descrivere la realtà di interesse. I costrutti si organizzano in schemi che descrivono organizzazione e struttura delle occorrenze (istanze).

Entità: classi di oggetti che hanno proprietà comuni ed esistenza autonoma nell'applicazione. Ha un nome che la identifica univocamente.

Relazioni o associazioni: legami logici tra due o più entità. Un'occorrenza di relazione è una ennupla costituita da occorrenze delle entità coinvolte. (Il libro vuole che si usino sostantivi e non verbi nelle relazioni). La relazione è una funzione, ovvero un sottinsieme del prodotto cartesiano. OCIO: ER non parla di tuple ripetute. Ci possono anche essere relazioni ricorsive, a volte con ruoli (eg precedente VS successivo).

Attributi: proprietà elementari di entità o relazioni, che servono all'applicazione. Un attributo associa a ciascuna occorrenza di entità o relazione un valore appartenente ad un dominio (valori ammissibili).

Attributo composto = raggruppato attributi che hanno attinenza.

Cardinalità delle relazioni: vanno specificate per ciascuna partecipazione di entità ad una relazione. Descrivono il numero minimo e massimo di occorrenze di relazione a cui una occorrenza di entità può partecipare (ovvero essere legata a occorrenze delle altre entità che partecipano).

Minima 0 = partecipazione opzionale

Minima 1 = partecipazione obbligatoria

Massima 1 = funzione (parziale se minima è 0)

Massima n = associata ad un numero arbitrario di occorrenze dell'altra entità

$(x, 1) - (x, 1)$ = relazioni uno a uno

$(x, 1) - (x, n)$ = relazioni uno a molti

$(x, n) - (x, n)$ = relazioni uno a molti

Cardinalità degli attributi: il numero minimo e massimo di valori per quell'attributo. Se è (1,1) viene omessa.

$(0, x)$ = opzionale

$(1, x)$ = obbligatorio

(x, n) = multivalore

Identificatori delle entità: sono gli attributi o le entità che rendono univocamente riconoscibili le varie occorrenze dell'entità.

Interno = chiave

Esterno = solo se l'entità partecipa con (1,1), e può coinvolgere anche altre entità, sempre solo se esse siano (1,1) e non si creino cicli di identificazione.

Generalizzazioni

Rappresentano legami logici tra un'entità detta padre e entità dette figlie, in cui le figlie sono un caso più particolare del padre, che rappresenta un caso generale.

Ogni occorrenza di una figlia è anche occorrenza di un padre. Ogni proprietà del padre è anche proprietà della figlia.

Classificazione delle generalizzazioni:

Totale = se ogni occorrenza del padre è almeno una occorrenza delle figlie (non esiste un padre "puro")

Parziale = esiste il padre puro

Esclusiva = un'occorrenza del padre è al max un'occorrenza di una figlia

Sovrapposta = un'occorrenza del padre può essere più occorrenze di figlie diverse

Le sovrapposte si possono trasformare in esclusive, aggiungendo delle figlie più specializzate.

6.3 Documentazione di schemi ER

6.3.1 Regole aziendali

Possono essere:

1. Descrizione di un concetto rilevante
2. Un vincolo di integrità
3. Una derivazione, ovvero come calcolare un attributo a partire dal valore di altri attributi

Descrizioni: linguaggio naturale

Vincoli di integrità: si usano asserzioni: <concetto> deve / non deve <espressione>

Derivazione: <concetto> si ottiene <operazione>

6.3.2 Tecniche di documentazione

Le regole aziendali possono essere rappresentate in un dizionario dei dati, composto da due tabelle:

1. entità dello schema con nome, descrizione, attributi e id
2. chiavi esterne

CAPITOLO 7 : Progettazione concettuale

Costruire un ER in base ai requisiti

7.1 Raccolta ed analisi dei requisiti

Sono espressi in linguaggio naturale e vanno quindi disambiguati chiedendo bene.

Le fonti sono:

- utenti
- documentazione esistente
- realizzazione preesistente

7.2 Criteri generali di rappresentazione

- Se un concetto descrive classi di oggetti autonome, è un'entità
- Se un concetto non ha vita autonoma, può essere un attributo
- Se ci sono due o più entità e un concetto che le associa, questo può essere una relazione
- Se ci sono concetti che derivano l'uno dall'altro, si può usare una generalizzazione

7.3 Strategie di Progetto

7.3.1 Strategia top-down

Si parte da uno schema iniziale con tutte le specifiche e pochi concetti astratti e lo si raffina successivamente con le primitive di trasformazione top-down.

T1 = da più entità a relazione tra queste entità

T2 = da entità a generalizzazione

T3 = da relazione a insieme di relazioni

T4 = da relazione a entità con relazioni

T5 = introduzione di attributi su entità

T6 = introduzione di attributi su relazioni

7.3.2 Strategia bottom-up

Si fondono concetti singoli fino ad ottenere lo schema generale.

T1 = genero un'entità

T2 = genero relazione tra 2 entità

T3 = genero generalizzazione

T4 = aggrego attributi ad una entità

T5 = aggrego attributi a una relazione

7.3.3 Strategia inside-out

Si individuano dei concetti importanti e si procede a macchia d'olio verso concetti vicini etc.

7.3.4 Strategia mista

Metto insieme tutte le cose per avere vantaggi. È l'unica effettivamente utilizzabile, perché essere fedeli al 100% ad una delle altre strategie è impossibile, in sistemi complessi.

7.4 Qualità di uno schema concettuale

Correttezza = quando usa correttamente il modello concettuale ER

Completezza = rappresenta tutte le informazioni e rende possibili tutte le operazioni

Leggibilità = comprensibile da più o meno tutti

Minimalità = non ci sono cose superflue, che pregiudicherebbero l'efficienza

CAPITOLO 8 : La progettazione logica

Parto da uno schema concettuale e cerco di produrre uno schema logico che rappresenti tutto quello che ER contiene. Occorre prima ristrutturare ER per ottimizzarlo (ed è una cosa indipendente dal modello logico di riferimento), e poi tradurlo nel modello logico, che nel nostro caso è quello relazionale.

La ristrutturazione prende ER e il carico applicativo e cerca di ottimizzare le cose.

8.2 Ristrutturazione di schemi ER

8.2.1 Analisi delle ridondanze

Una ridondanza è un dato che può essere derivato da altri dati già presenti:

- attributo che deriva da altri attributi della stessa entità
- attributo che deriva da attributi di altre entità
- attributo che sarebbe un conteggio di occorrenze
- relazioni derivabili dalla composizione di altre relazioni mediante cicli

Dati derivati:

PRO = non si accede a tante tabelle per avere il valore di quell'attributo

CONTRO = devo usare più memoria e tempo per fare i conti

8.2.2 Eliminazione delle generalizzazioni

Accorpamento delle figlie nel padre: metto gli attributi delle figlie nel padre, ed un selettore per poterle identificare (e magari regole aziendali per evitare che un attributo abbia un valore quando non dovrebbe averne). Conviene se ho poche distinzioni nelle figlie.

Accorpamento del padre nelle figlie: tutti gli attributi del padre vengono passati ad ogni entità delle figlie. Si può usare solo se la generalizzazione è totale, e in questo caso si risparmia memoria.

Sostituzione della generalizzazione con associazioni: si trasforma la generalizzazione in associazioni (1,1) che legano il padre con le figlie, e non si trasferiscono attributi da padre a figlio, ma occorre precisare dei vincoli. Si risparmia memoria perché non ci sono valori nulli, ma aumentano gli accessi. Meglio se la generalizzazione è totale.

8.2.3 Partizionamento di concetti

Partizionamento di entità

- **Decomposizione verticale:** sugli attributi. Da un'entità ne ricavo 2 legate da una relazione (1,1) per entrambe, e distinguo i dati di interesse per 2 scopi diversi: eg impiegato => anagrafica VS dati lavorativi.
- **Decomposizione orizzontale:** sulle occorrenze, ovvero una generalizzazione, logicamente parlando.

Eliminazione di attributi multivalore

Il modello relazionale non permette di rappresentare direttamente questa caratteristica dell'ER, e quindi si rende necessaria, in genere, una nuova tabella.

Accorpamento di entità

Due entità legate con (1,1) vengono fuse in un'unica entità, con gli attributi di entrambe. Raro farle su una (1,n), praticamente mai su (n,n).

Anche le associazioni possono essere partizionate o accorpate, ma in genere sto mestiere si fa a livello di progettazione fisica.

8.2.4 Scelta degli identificatori principali

- NO attributi con valori nulli
- pochi attributi sono meglio di tanti (meno accessi)
- interno meglio di esterno
- meglio utilizzato da tante operazioni => posso dover creare un indice apposito

8.3 Traduzione verso il modello relazionale

Lo schema logico deve essere equivalente, cioè rappresentare le stesse informazioni.

8.3.1 Entità e associazioni molti a molti

- Entità => relazione
- Associazione => relazione con attributi suoi e id delle entità che coinvolge, e ciò le fa da chiave.

8.3.2 Associazioni uno a molti

Accorpo in una tabella, se la cardinalità minima è 1 (la max è 1 cmq)

8.3.3 Entità con identificatore esterno

L'id esterno toglie la necessità di avere l'associazione tra le 2 entità.

8.3.4 Associazioni uno a uno

Se sono obbligatorie, posso avere 2 entità con integrità referenziale tra di loro.

Posso anche farle diventare 1 tabella unica, perché l'associazione (1,1) è una funzione biunivoca. Ma ciò non va bene perché se sono arrivato a distinguere concetti diversi nell'ER, per qualche motivo, non è opportuno cancellarli nel modello logico.