Appunti Basi di Dati

Luca Seggiani

24 Aprile 2024

1 Strategie di progetto

Possiamo definire alcuni tipi di strategie di progetto:

- **Top-down**
  Si parte da uno schema iniziale e generale (top) che viene poi definito nei dettagli fino allo schema definitivo (down), muovendosi così, figurativamente, dall'alto verso il basso (zip!). La raffinazione dettagli avviene attraverso l'applicazione di primitive:

  Primitive di raffinamento

  - Da entità a associazione tra entità;
  - Da entità a generalizzazione;
  - Da associazione a insiemi di associazioni;
  - Da associazione a entità con associazioni;
  - Introduzione di attributi su entità e associazioni.

  Dove associazione è sinonimo di relationship.

- **Bottom-up**
  Si parte dalle specifiche delle componenti minime dello schema: queste specifiche vengono poi integrate fra di loro a formare lo schema completo. Come prima, figurativamente, dal basso verso l'alto. Questa integrazione avviene attraverso le primitive di trasformazione (o generalizzazione).

  Primitive di trasformazione

  - Generazione di entità;
  - Generazione di associazione;

  - Generazione di generalizzazione.

  Come sopra.

- **Inside-out**
  La progettazione inside-out è sostanzialmente una generalizzazione della bottom-down, dove si parte da strutture atomiche di complessità minore fino a schemi più complessi e sempre più vicini al risultato finale.

Nella pratica, una strategia non è mai puramente top-down o bottom-up, ma è più una **strategia mista**, che può essere sintetizzata come:

- Individuazione dei concetti principali e realizzazione di uno schema scheletro (un semplice schema concettuale che racchiude i dettagli più importanti della realtà da modellare);
- Decomposizione dello schema scheletro;
- Raffinazione, espansione e integrazione delle componenti dello schema.

Possiamo definire questa metodologia nei dettagli.

- **Analisi dei requisiti**
  - Analisi dei requisiti ed eliminazione delle ambiguità;
  - Costruzione di un glossario di termini;
  - Raggruppamento dei requisiti in insiemi omogenei.
- **Passo base**
  - Definizione uno schema scheletro con i concetti più rilevanti.
- **Passo iterativo**
  - Raffinazione dei concetti presenti sulla base delle loro specifiche;
  - Aggiunta di concetti necessari a descrivere specifiche non implementate.
- **Analisi di qualità**
  - Verifica della qualità dello schema e seguente modifica se necessaria. La qualità di uno schema concettuale può essere riassunta nella sua:

    - **Correttezza**
    - **Completezza**
    - **Leggibilità**
    - **Minimalità**

2 Progettazione logica

Giungiamo adesso alla progettazione logica del nostro schema. L'obiettivo della progettazione logica è quello di tradurre lo schema concettuale in uno schema logico che rappresenti i dati in modo corretto ed efficiente. Prendiamo quindi in ingresso lo schema concettuale, il modello logico usato e una serie di informazioni sul **carico applicativo**, ovvero lo sforzo a cui sarà sottoposto il sistema; in uscita avremo uno schema logico ben definito con relativa documentazione.

Valutazione delle prestazioni
La valutazione delle prestazioni può essere effettuata attraverso determinati indicatori, ovvero parametri che caratterizzano le prestazioni. I più importanti sono

- **Spazio**: numero di occorrenze previste;
- **Tempo**: numero di occorrenze (o relationship) visitate per portare a termine un'operazione.

Il calcolo degli indicatori di tempo può essere effettuato attraverso una tavola degli accessi: la tavola degli accessi relativa a una data operzione si ricava dallo schema di navigazione, costruito sul diagramma ER.

Ristrutturazione dello schema ER
A volte potrebbe essere necessario ristrutturare lo schema ER: ciò permette di semplificarne la traduzione e ottimizzare le prestazioni del sistema. Notiamo che uno schema ER ristrutturato non è più uno schema concettuale nel senso stretto del termine. La ristrutturazione si effettua attraverso i seguenti passaggi:

- **Analisi delle ridondanze**
  Una ridondanza in uno schema ER è un'informazione significativa ma derivabile da altre. Nella fase di progettazione logica si decide se eliminare le ridondanze eventualmente presenti nello schema oppure mantenerle (se non addirittura introdurne di nuove). Il vantaggio delle ridondanze è la semplificazione delle interrogazioni; gli svantaggi sono il maggiore spazio richiesto e l'appesantimento delle operazioni di

aggiornamento. Le principali forme di ridondanza in uno schema ER sono:

  - **Attributi derivabili**
    Attributi che si possono derivare da attributi o della stessa entità o di entità diverse (o relationship).
  - **Relationship derivabili**
    Relationship che si possono derivare dalla composizione di altre relationship (e.g. cicli).

- **Eliminazione delle generalizzazioni**
  Il modello relazionale non può rappresentare direttamente le generalizzazioni: dispone solamente di entità (relazioni) e relationship (dipendenza). Per questo motivo è necessario eliminare le gerarchie, sostituendole a loro volta con entità e relationship. Le metodologie possibili sono:

  - **Accorpamento delle figlie** della generalizzazione nel genitore, ovvero l'eliminazione della generalizzazione figlia, che verrà trasformata in una relationship ad entità la cui esistenza o non è segnalata da un certo attributo ("tipo"). Si noti che l'entità di partenza mantiene i suoi attributi;
  - **Accorpamento del genitore** della generalizzazione sulle figlie, ovvero un processo simile al precedente in cui si trasforma la gerachia in una serie di relationship. In questo caso gli attributi vanno a ricadere sulle figlie (si "accorpa" sulle figlie), e il genitore può essere eliminato direttamente;
  - **Sostituzione** della generalizzazione con relationship, ovvero la trasformazione diretta in relationship, dove gli attributi dell'entità genitrici e generalizzate restano alle loro entità d'appartenenza, e la loro gerarchia viene trasformata in relationship con cardinalità (0,1) $\rightarrow$ (1,1).

  La scelta fra le alternative possibili di eliminazione di generalizzazione può essere fatta sulla base della tabella degli accessi, seguendo alcune linee guida generali:

  - L'accorpamento delle figlie conviene se gli accessi al padre e alle figlie sono contestuali;
  - L'accorpamento del genitore conviene se gli accessi al padre e alle figlie sono differenti;

  - La sostituzione conviene se gli accessi alle entità figlie sono separati da quelli al padre.

  Notiamo poi che non sono escluse soluzioni ibride, sopratutto nel caso di gerarchie a più livelli.

- **Partizionamento/accorpamento di entità e relationship**
  Passiamo adesso alle ristrutturazioni fatte per rendere più efficienti le operazioni in base al principio di riduzione degli accessi. Questo si può ottenere:

  - **Separando gli attributi** di un concetto a cui si accede separatamente;
  - **Raggruppando attributi** di concetti diversi a cui si accede insieme.

  Considereremo, per semplicità, che ad ogni accesso si legge sempre l'intera informazione. Le casistiche principali saranno:

  - **Partizionamento verticale** di entità: il partizionamento verticale consiste nel dividere una data entità in due (sotto)entità, con un'opportuna relationship fra di esse. Ognuna delle due entità è in corrispondenza biunivoca con l'altra (cardinalità (1,1)) con chiave esterna corrispondente alla relationship per l'entità che non si prende la chiave interna.
  - **Partizionamento orizzontale** di relationship: il partizionamento orizzontale consiste nel dividere una data relationship dotata di attributi in più relationship corrispondenti a diversi valori di un attributo. Ad esempio, la divisione di una relationship d'appartenenza a due relationship di "appartenza corrente" e "appartenenza passata" in termini temporali.
  - **Eliminazione di attributi multivalore**: l'eliminazione di attributi multivalore viene incontro all'impossibilità di rappresentare attributi multivalore nel modello relazionale. Si trasformano quindi dati attributi in un'entità a sé stante che sarà collegata all'entità di partenza con una relazione di cardinalità (1,N) $\rightarrow$ (1,1) (o (1,N)?).
  - **Accorpamento di entità e relationship**: l'accorpamento di entità e relationship è l'esatto opposto del partizionamento verticale: due entità collegate da una relationship di cardinalità (1,1) $\rightarrow$

(0,1) potranno tranquillamente essere accorpate in un'unica entità con attributi a cardinalità (0,1) corrispondenti agli ex attributi della seconda entità.

- **Scelta di identificatori primari** La scelta degli identificatori primari è indispensabile alla traduzione nel modello relazionale, dove esiste il concetto di chiave. I criteri da adottare sono:
  - L'assenza di opzionalità;
  - La semplicità;
  - Utilizzo nelle operazioni più frequenti o importanti

  Nel caso nessuno degli identificatori soddisfi i requisiti necessari può essere necessario, sebbene sia poco elegante, introdurre nuovi attributi (codici) generati appositamente, magari in maniera incrementale.

Traduzione verso il modello relazionale
Approcciando la traduzione nel modello relazionale dello schema logico, dovremmo pensare di trasformare entità in relazioni sugli stessi attributi, e relationship in relazioni sugli identificatori delle entità coinvolte.