

Appunti Basi di Dati

Luca Seggiani

10 Aprile 2024

1 Modellazione e progettazione concettuale

La definizione di schemi adeguati per le basi di dati richiede metodologie precise per la modellazione accurata della realtà che ci interessa. Bisogna tenere a mente che:

- Non conviene concentrarsi subito sui dettagli;
- Conviene invece stabilire subito interdipendenze fra relazioni;
- Il modello relazionale sarà rigido una volta ultimato.

In generale si può dire che esiste sempre (e spesso è unico) uno schema che modella accuratamente e nel modo più semplice possibile una certa realtà di interesse. La progettazione di basi di dati è solo una fase dello sviluppo di un sistema informativo (vedere a riguardo le prime lezioni di teoria). Bisogna quindi tenere conto di:

- **Ciclo di vita** (*lifecycle*) del sistema informativo, ovvero l'insieme delle attività svolte da analisti, progettisti e utenti nello sviluppo e nell'uso del sistema informativo. Questa attività è iterativa, e quindi ciclica.

I passi ciclo di vita dovranno essere ben definiti attraverso linguaggi e modelli prestabili. Per le basi di dati, in particolare, conviene adottare modelli di facile utilizzo, che consentano la decomposizione delle attività in fasi (e/o livelli) distinti, e di utilizzare strategie e criteri di scelta nei vari passaggi.

Modello a cascata

Il modello a cascata (*waterfall model*) le fasi sono sequenzialmente ordinate, etichettate, e non ripetibili. In ordine, esse sono le seguenti:

1. **Studio di fattibilità:** definizione di costi e priorità della produzione;

2. **Raccolta e analisi dei requisiti:** studio delle proprietà del sistema che andranno implementate;
3. **Progettazione:** progettazione di strutture dati e funzioni;
4. **Realizzazione:** implementazione effettiva del codice;
5. **Validazione e collaudo:** sperimentazione del prodotto;
6. **Funzionamento:** il sistema diventa operativo in produzione (*shipping*).

Vediamo alcune fasi nel dettaglio.

2 Raccolta e analisi dei requisiti

Questa fase può essere, a sua volta, divisa in due sotto-fasi:

- **Acquisizione dei requisiti:** il reperimento dei requisiti è un'attività non standardizzata. Esistono più modalità:
 - Direttamente dagli utenti:
 - * Interviste, focus group, recensioni, ecc...
 - * Documentazioni apposite;
 - Attraverso documentazioni preesistenti:
 - * Normative (legislazioni, regolamenti di settore);
 - * Regolamenti interni, procedure aziendali;
 - * Realizzazioni preesistenti.

Esistono linguaggi per definire requisiti (*UML*).

- **Analisi dei requisiti:** si analizzano i requisiti raccolti, spesso nella prospettiva di successive acquisizioni.

Interazione con gli utenti

Le problematiche dell'interazione con gli utenti possono essere:

- Utenti diversi danno risposte diverse;
- Utenti a livello più alto hanno spesso una visione più ampia ma meno dettagliata;
- Spesso l'acquisizione di requisiti avviene per raffinazione.

Conviene quindi:

- Effettuare spesso verifiche di comprensione e coerenza;
- Verificare anche attraverso esempi (soprattutto nei casi limite);
- Richiedere definizioni e classificazioni chiare e specifiche;
- Separare gli aspetti essenziali da quelli marginali (*ranking*).

Interazione con gli utenti tramite documentazione

E' opportuno seguire alcune linee guida generali, assicurando:

- Standardizzazione della struttura delle frasi;
- Separazione delle frasi riguardanti dati da quelle riguardanti funzioni;
- Organizzazione di termini e concetti:
 - Unificazione di termini (eliminazione di sinonimi);
 - Esplicitazione del riferimento fra termini;
- Organizzazione delle frasi per concetti.

3 Progettazione

La progettazione è una fase del ciclo di vita. Per un sistema software la progettazione si divide effettivamente in:

- Progettazione dei dati;
- Progettazione delle applicazioni.

Progettazione per astrazione

Come in tutte le applicazioni informatiche, abbiamo visto che è necessario progettare per livelli successivi di astrazione:

- **Livello concettuale:** esprime i requisiti di un sistema in una descrizione adatta all'analisi da punti di vista esterni
- **Livello logico:** evidenzia l'organizzazione dei dati dal punto di vista del loro contenuto informativo, descrivendo la struttura dei record e le loro interdipendenze.

- **Livello fisico:** si concentra sulla base di dati vista come un insieme di blocchi fisici sul disco, e riguarda quindi l'allocazione dei dati e le modalità di memorizzazione.

Con riferimento a quanto detto sugli schemi logici avremo quindi che la progettazione si divide in:

- **Progettazione concettuale**, parte dai requisiti individuati della base e produce uno schema concettuale;
- **Progettazione logica**, parte dallo schema concettuale e produce uno schema logico;
- **Progettazione fisica**, parta dallo schema logico e produce lo schema fisico finale.

Come lo era stato il modello a cascata, la progettazione per astrazione è composta da fasi sequenzialmente ordinate che vanno eseguite strettamente in ordine.

Modello dei dati

Il modello dei dati è l'insieme dei costrutti utilizzati per organizzare i dati di interesse e definirne la dinamica. Componente fondamentale del modello sono i meccanismi di strutturazione (costruttori di tipi). Come per i linguaggi di programmazione comuni, esistono meccanismi che permettono di definire nuovi tipi. Ogni modello dei dati prevede alcuni costruttori: ad esempio il modello relazionale prevede un costruttore relazionale che permette di definire insiemi di record omogenei. Per riassumere in breve, in ogni base di dati si ha:

- Lo **schema**, invariante nel tempo, che ne descrive la struttura. Si notino inoltre le **intestazioni** delle tabelle (previste nel modello relazionale).
- L'**istanza**, i valori effettivi assunti dalla base di dati in un dato momento. Nel modello relazionale rappresenta il **corpo** di ciascuna tabella.

Modello concettuale Entity-Relationship

Il modello concettuale che utilizzeremo sarà quello **Entity-Relationship (ER)**. Si noti che la parola *relationship*, sebbene abbia lo stesso significato in lingua inglese, non si riferisce al concetto matematico di relazione che sta alla base del modello relazionale. Per questo motivo useremo sempre il termine inglese per descrivere le relazioni del modello entity-relationship, in modo da distinguere dalle relazioni del modello relazionale. Il modello entity-relationship viene sviluppato da P.P. Chen nel 1976, ed è oggi una

delle metodologie più affermate nel campo della progettazione dei sistemi informatici, anche se in un'accezione leggermente diversa da quella in cui era stato concepito inizialmente.

I suoi costrutti base sono:

- Entità
- Relationship
- Attributi

Entità

Un'entità è una classe di oggetti dell'applicazione d'interesse con proprietà comuni e esistenza autonoma. Un'occorrenza (o istanza) di entità è un elemento della classe (un'elemento, non i dati ad esso legati!). Ogni entità deve avere un nome che la identifica univocamente nello schema. Graficamente è rappresentato da una scatola.

Relationship

Una relationship è un legame logico fra due o più entità, rilevante nell'applicazione d'interesse. Può essere chiamata anche relazione (vedi sopra), correlazione o associazione. Ogni relationship, come per le entità, ha un nome che la identifica univocamente nello schema. Graficamente è rappresentata da una losanga.

Vediamo allora di definire il concetto di occorrenza di relationship, più complesso di quello di occorrenza di entità:

- Un'occorrenza di **relationship binaria** è una coppia di occorrenza di entità, una per ciascuna entità coinvolta.
- Una occorrenza di una **relationship n-aria** è una n-upla di occorenze di entità, una per ciascuna delle n entità coinvolte.

Nell'ambito di una relationship non ci possono essere occorrenze (né coppie né n-uple) ripetute.

Attributo

L'attributo è una proprietà elementare di un'entità o di una relationship, che ci interessa ai fini dell'applicazione d'interesse. Associa a ogni occorrenza di entità o relationship un valore appartenente ad un dominio, il cosiddetto dominio dell'attributo.

Attributo composito

Gli attributi compositi raggruppano attributi di una medesima entità o relationship che presentano affinità nel loro significato o uso (e.g. giorno, mese, anno si compone in data, via, numero civico, CAP in indirizzo, ecc...).