Appunti Basi di Dati (Teoria)

Luca Seggiani

7 Marzo 2024

1 Modelli dei Dati

Modelli logici
Adottati nei DBMS esistenti per l'organizzazione dei dati, indipendenti dalle strutture fisiche. Esempi: relazionale, reticolare, gerarchico, a oggetti, basato su XML.

Modelli concettuali
Permettono di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti del mondo reale. Sono utilizzati nelle fasi preliminari di progettazione.
Esempio: Entity-Relationship (ER).

Possiamo dire che l'utente si interfaccia con lo schema logico di un database, dettato dal modello logico adottato. A livello fisico accade quanto definito nello schema interno, ovvero la parte di effettiva implementazione del DBMS.

Schema logico
Lo schema logico è la descrizione della base di dati nel modello logico, quindi la struttura di tabelle, ecc... L'utente che sviluppa il database si interfaccia con lo schema logico, usufruendo delle sue astrazioni.

Schema interno
Lo schema interno è l'implementazione dello schema logico, ed è noto solo a chi implementa il DBMS.

2 Linguaggi per Basi di Dati

I DBMS dispongono di vari linguaggi e interfacce per la definizione di schemi, modifica e lettura dei dati, e formmulazione di query. Possiamo avere:

- Linguaggi testuali interattivi (SQL)

- Comandi (SQL) immersi in un linguaggio ospite (Java, C++, ...)
- Interfaccie grafiche (Access)

Si può fare l'ulteriore distinzione:

- **Data definition language** (DDL): per la definizione di schemi (logici e fisici)
- **Data manipulation language** (DML): per l'interrogazione e l'aggiornamento di istanze di basi di dati.

3 Architettura a tre livelli per DBMS

Possiamo complicare le cose introducendo, oltre allo schema interno e lo schema logico, un'ulteriore schema, lo schema esterno, attraverso cui permetteremo agli utenti di interfacciarsi con la base di dati a livello ancora più alto (astratto).
Lo schema esterno immplementa fondamentalmente viste parziali (magari solo alcune tabelle) di database. Sarà in ogni caso importante stabilire:

- Indipendenza fisica: il livello logico e quello esterno sono indipendenti da quello fisico, come nel paradigma dell'astrazione procedurale, la realizzazione dello schema fisico può variare senza che debbano essere attuate modifiche dello schema logico.
- Indipendenza logica: il livello esterno è poi indipendente da quello logico, aggiunte o modifiche alle viste non richiedono modifiche al livello logico, e viceversa le modifiche dello schema logico lasciano inalterato lo schema esterno.

4 Attori

Possiamo adesso stabilire quali sono gli "attori" che implementeranno, lavoreranno su ed interagiranno con il DBMS:

- Progettisti e realizzatori di DBMS
- Progettisti della base di dati e suoi amministratori
- Progettisti e programmatori di applicazioni
- Utenti:

- Utenti finali: eseguono applicazioni predefinite:
- Utenti casuali: eseguono operazioni noon previste a priori, usando linguaggi interattivi (SQL)

5 Modello relazionale

Vale la pena riportare i 3 modelli logici storici, tra cui figura il modello relazionale che andremo a studiare:

gerarchico, reticolare, relazionale

Più recentemente si è poi diffuso il paradigma ad oggetti, basato su XML, anche detto NoSQL (viene principalmente usato su database di dimensioni particolarmente grandi).

I modelli gerarchici e reticolari non sono molto utilizzati per un particolare difetto: la gestione di relazioni fra dati non viene gestita in maniera efficiente (si usa il meccanismo dei riferimenti, che sono però fin troppo dipendenti dalla struttura fisica adottata). Nel modello relazionale, invece, tutto è basato sui valori, completamente slegati alle specificità della struttura fisica. Gli stessi riferimenti, o più propriamente relazioni, sono basati su valori condivisi fra più tabelle.

Tabelle
La tabella è l'entità fondamentale di un database relazionale. Come già visto prima, una tabella contiene righe (record) e colonne (attributi), che rappresentano in un qualsiasi momento un istanza dello schema logico adottato.
Relazioni
Le relazioni del database relazionale si basano sul concetto matematico di relazione. Poniamo ad esempio due insiemi:

$$D_1 = (a, b)$$

$$D_2 = (x, y, z)$$

e il loro prodotto cartesiano:

$$D_1 \times D_2 = ((a, x), (a, y), (a, z), (b, x), ...)$$

una certa relazione R è:

$$R \subseteq D_1 \times D_2$$

sottoinsieme del loro prodotto cartesiano. Formalmente:
Dati n insiemi (anche non distinti) $D_1, D_2, ..., D_n$, e definito su di essi un

prodotto cartesiano $D_1 \times D_2 \times ... \times D_n$, ovvero l'insieme di n-uple ordinate $(d_1, d_2, ..., d_n)$ con $d_1 \in D_1, d_2 \in D_2, ..., d_n \in D_n$, una relazione è un sottoinsieme del prodotto cartesiano fra gli insiemi.
Non esiste ordinamento fra le n-uple, e non esistono n-uple uguali. Gli insiemi $D_1, D_2, ..., D_n$ sono detti domini della relazione.

Struttura non posizionale
Dando un certo nome ai domini della relazione, possiamo passare da quella che è effettivamente una struttura posizionale (le n-uple sono ordinate), ad una struttura dove le posizioni dei domini (attributi) non sono più rilevanti.

Riferimenti fra relazioni
I riferimenti fra relazioni diverse sono rappresentati da valori dei domini che compaiono nelle n-uple.

Definizioni formali
Abbiamo adesso tutti gli strumenti necessari a definire formalmente il modello relazionale:

- **Schema di relazione**: simbolo R detto nome della relazione e un insieme di attributi $X = (A_1, ..., A_2)$ solitamente indicato con $R(X)$. A ciascun attributo X è associato un dominio. Gli attributi non possono essere relazioni.

- **Schema di base di dati**: un insieme di schemi di relazione con nomi diversi, solitamente indicato come $R = (R_1(X_1), ..., R_n(X_m))$.

N-uple
Una n-upla o tupla su insieme di attributi X è una funzione che associa a ciascun attributo $A \in X$ un elemento, o valore, nel dominio di A. Il simbolo $t[A]$ denota il valore della n-upla t sull'attributo A. Il simbolo $t[Y]$, con $Y \subseteq X$, denota il valore della n-upla t sull'insieme di attributi Y. N.B.: le n-uple non sono ordinate!

Definiamo allora le istanze di relazioni (e quindi di database):

- **Istanza di relazione**: su uno schema $R(X)$, un insieme r di n-uple su X

- **Istanza di base di dati**: su uno schema $R = (R_1(X_1), ..., R_n(X_m))$, un insieme di relazioni $r = (r_1, ..., r_2)$ dove ogni r_i è una relazione sullo schema $R_i(X_i)$

Graficamente, una n-upla è una riga della tabella, e un istanza l'insieme di tutte le sue righe.

Informazione incompleta

Il modello relazionale impone ai dati una struttura rigida, ma non è detto che i dati del mondo reale aderiscano sempre a questa struttura. Si introduce allora la parola chiave NULL, per evitare di usare valori particolari del dominio (0, ecc...), che potrebbero diventare significativi, o la cui gestione è comunque abbastanza complicata. Il valore di un certo attributo A, ovvero $t[A]$, potrà quindi appartenere al dominio D_A oppure essere il valore NULL. Si possono inoltre imporre determinate restrizioni sulla presenza dei valori nulli in una data relazione. Un valore NULL può modellizzare almeno 3 casi differenti:

- valore sconosciuto
- valore inesistente
- valore senza informazione

N.B.: Il DBMS non fa alcuna distinzione fra valori NULL!
La limitazione sui valori che possono o non possono essere NULL torna utile ad esempio nel caso della definizione di chiavi primarie di record, che devono essere fra di loro diverse ma comunque mai nulle.