

Appunti di Basi di Dati

4 ottobre 2022

Rosso Carlo

Contents

1	introduzione	2
2	Algebra relazionale	2
2.1	vincoli intrarelazione	3
2.2	vincoli interrelazionali	3
2.3	operatori su relazioni	3
2.4	viste	4

1 introduzione

Def. 1.1 (base di dati) *insieme organizzato, persistente e condiviso di dati utilizzati per lo svolgimento delle attività automatizzate di un'organizzazione.*

Def. 1.2 (dato) *ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione.*

Def. 1.3 (informazione) *notizia, dato o elemento che consente di avere conoscenze più o meno esatte di fatti, situazioni, modi di essere.*

Elaborazione dei dati permette di estrarre delle informazioni dai dati.

Def. 1.4 (DataBase Management System (DBMS)) *Le basi di dati sono gestite da un DBMS. Una base di dati è grande, persistente e condivisa. Ovvero, grandi: il limite deve essere solo quello fisico dei dispositivi; persistenti: indipendente dalle singole esecuzioni dei programmi; condivise: applicazioni e utenti diversi accedono ad una porzione (sovrapponibile) delle basi di dati).*

Un DBMS garantisce alla base di dati: privacy, affidabilità, efficienza, efficacia.

La proprietà della condivisione che caratterizza i database ha alcuni problemi che devono essere risolti:

- ridondanza: dati duplicati;
- incoerenza: le versioni possono non coincidere;
- privacy: i fruitori delle basi di dati devono accedere ai dati per cui sono autorizzati.

Le soluzioni sono le seguenti:

- indipendenza delle operazioni: sono permesse attività diverse solo su dati diversi; dunque, sono previsti meccanismi di autorizzazione, per garantire la sequenzialità delle operazioni;
- accessi di più utenti a dati condivisi: meccanismi di controllo della concorrenza e talvolta è impiegato l'uso di lock e semafori.

I DBMS garantiscono affidabilità sfruttando le transazioni.

Def. 1.5 (transazioni) *insieme di operazioni, da considerare indivisibile, corrette anche in presenza di concorrenza e con effetti definitivi.*

Def. 1.6 (schema) *lo schema è quasi un'invariante nel tempo. Si tratta della descrizione delle tabelle contenute nel database.*

Def. 1.7 (istanza) *è la rappresentazione dei dati in un dato momento, nei modi indicati attraverso lo schema. L'istanza è soggetta a variazioni nel tempo.*

Un contributo all'efficacia dei DBMS è la disponibilità di vari linguaggi e interfacce. Per esempio il linguaggio testuale interattivo SQL, che si impegna a rappresentare l'algebra relazionale. In realtà i comandi SQL possono essere immersi in un linguaggio ospite, come ad esempio C++. Oppure si può gestire un DBMS attraverso un'interfaccia amichevole.

SQL svolge una doppia funzione: è un data manipulation language (DML), per cui, permette di interrogare e di aggiornare istanze di basi di dati; è un data definition language (DDL), per cui, permette di definire e modificare gli schemi.

2 Algebra relazionale

Def. 2.1 (relazione) *una relazione è un insieme di tuple (attributi), ciascuna delle quali è un insieme di valori, uno per ogni attributo. Gli attributi non hanno ordine, gli elementi dell'insieme non hanno ordine. Per cui (in un database), l'ordinamento tra le righe è irrilevante; l'ordinamento tra le colonne è irrilevante; le righe sono diverse tra loro; le intestazioni delle colonne sono diverse tra loro; i valori di ogni colonna sono definiti su domini omogenei.*

Pro	Contro
condividere i dati in un'organizzazione	ne vale la pena solo per grandi organizzazioni
gestione centralizzata, easy to scale	non utilizzare senza tanti utenti
servizi integrati	non utilizzare senza accessi concorrenti e stabili
riduzione di ridondanze e inconsistenze	
indipendenza dei dati per la manutenzione e lo sviluppo	

Table 1: Pro e contro dei DBMS

Talvolta, l'attributo di un'istanza potrebbe non avere alcun valore. In questo caso si ricorre all'utilizzo di valori speciali, che indicheranno l'assenza di valore. Questo permette di imporre restrizioni sulla presenza di valori nulli. Il valore nullo può avere tre origini, nasconde uno di tre significati: il valore è sconosciuto; il valore non esiste; il valore non ha informazione intrinseca.

Def. 2.2 (sintatticamente corrette) *tutte le tuple sono diverse tra loro.*

Def. 2.3 (semanticamente corrette) *tutte le tuple rispettano le restrizioni imposte. I valori assunti hanno senso per l'applicazione di interesse.*

2.1 vincoli intrarelazione

Def. 2.4 (non nullo) *Data r di $R = A_1, \dots, A_n$, e $A_i \in R$, allora si dice che A_i è non nullo in r se e solo se per ogni $t \in r$, $t[A_i]$ è definita.*

Def. 2.5 (unicità) *Sia $\bar{R} = A_{i_1}, \dots, A_{i_k} \subset R$ un sottoinsieme non vuoto ($k \geq 1$). Allora si dice che \bar{R} è unico in r se e solo se per ogni $t_1, t_2 \in r$, $t_1[\bar{R}], t_2[\bar{R}]$ definiti abbiamo $t_1[\bar{R}] = t_2[\bar{R}] \Rightarrow t_1 = t_2$.*

Def. 2.6 (superchiave) *\bar{R} si un'istanza r di R è un sottoinsieme $\bar{R} \subset R$ che soddisfa il vincolo di unicità su r .*

Def. 2.7 (chiave) *\bar{R} di un'istanza di R è una superchiave $\bar{R} \subset R$ per r tale che $\forall \tilde{R} \subset \bar{R}$, \tilde{R} non è una superchiave per R (minimalità).*

Def. 2.8 (chiave primaria (primary key, pk)) *\bar{R} di r su R è una chiave di r per cui vale non nulla su ogni attributo di \bar{R} in r .*

Def. 2.9 (schema di relazione) *Lo schema di una relazione R è costituito da una relazione R e un insieme di vincoli V_1, \dots, V_c intrarelazionali su R .*

Def. 2.10 (istanza (valida) di uno schema) *r di R è un'istanza (valida) di R se e solo se r soddisfa V_1, \dots, V_c .*

2.2 vincoli interrelazionali

Def. 2.11 (chiave esterna (foreign key)) *Date 2 relazioni $R_s = A_1, \dots, A_{m_a}$, $R_t = B_1, \dots, B_{m_b}$, una chiave esterna da R_s verso R_t è un insieme finito di coppie $FK = \{(A_{i_1}, B_{j_1}), \dots, (A_{i_k}, B_{j_k})\}$, dove B_{j_i} , tale che $x \in [1, \dots, k]$, sono tutti distinti e $Dom(A_{i_x}) = Dom(B_{j_x})$ per ogni $x \in [1, \dots, k]$.*

Def. 2.12 (vincolo di chiave esterna) *$R_s = A_1, \dots, A_{m_a}$, $R_t = B_1, \dots, B_{m_b}$, $FK = \{(A_{i_1}, B_{j_1}), \dots, (A_{i_k}, B_{j_k})\}$, da R_s verso R_t , r di R_s , \bar{r} di R_t , allora si dice che r soddisfa il vincolo di chiave esterna FK se e solo se:*

1. B_{j_1}, \dots, B_{j_k} è pk in \bar{r} ;
2. sia $\forall t \in r$, tale che $t[A_{i_1}], \dots, t[A_{i_k}]$ definito, allora $\exists \bar{t} \in \bar{r}$ tale che $t[A_{i_1}], \dots, t[A_{i_k}] = \bar{t}[B_{j_1}], \dots, \bar{t}[B_{j_k}]$.

2.3 operatori su relazioni

L'insieme degli operatori su relazioni producono relazioni e possono essere composti. Operatori insiemistici, le relazioni sono insiemi, il risultato sono altre relazioni è possibile applicare unione, intersezione e differenza solo a relazioni definite sugli stessi attributi.

ridenominazione $[\rho]$ modifica lo schema lasciando inalterata l'istanza dell'operando. La ridenominazione è un operatore unario.

selezione $[\sigma]$ il codominio è un sottoinsieme del dominio. La selezione è un operatore unario. Sono tolte delle tuple.

proiezione $[\pi]$ il codominio è un sottoinsieme del dominio. La proiezione è un operatore unario. Sono tolti degli attributi. Notiamo che se X è una superchiave di R , allora $\pi_X(R)$ ha cardinalità uguale a quella di R .

join $[\bowtie]$ operatore binario su due relazioni A e B . Il risultato è l'unione degli attributi degli operandi; in particolare, il risultato sono le tuple costruite con $eA \times B$ mantenendo quelle con valori uguali su attributi uguali.

Date due relazioni $R_1(X_1)$, $R_2(X_2)$, $R_1 \bowtie R_2$ è una relazione su X_1X_2 , ovvero $X_1 \cup X_2$, del tipo

$$R_1 \bowtie R_2 = \{t \in X_1X_2 \mid t[X_1] \in R_1, t[X_2] \in R_2\}. \quad (1)$$

Θ -join si tratta del prodotto cartesiano tra le relazioni operando. Diventa utile se combinato alla selezione, infatti si impone una condizione che deve essere rispettata. In effetti il Θ -join è un join a cui è applicata la selezione:

$$\sigma_{condizione}(R_1 \bowtie R_2) = R_1 \bowtie_{condizione} R_2. \quad (2)$$

In algebra, due espressioni sono equivalenti se producono lo stesso risultato qualunque sia l'istanza attuale delle relazioni della base di dati.

I DBMS usano regole di equivalenza per ottimizzare le interrogazioni. L'obiettivo delle ottimizzazioni è quello di minimizzare il numero di tuple e di attributi nelle relazioni intermedie.

In SQL esistono forme apposite per riferirsi ai valori nulli: **IS NULL** e **IS NOT NULL**.

2.4 viste

Le viste sono rappresentazioni diverse per gli stessi dati; ovvero, sono relazioni il cui contenuto è funzione del contenuto di altre relazioni.

Alle viste si contrappongono le relazioni di base, che hanno contenuto autonomo.

viste materializzate si tratta di relazioni derivate (in quanto si tratta di viste) memorizzate nella base di dati. Il pro delle viste materializzate è che non c'è bisogno di ricalcolare la vista ogni volta che viene richiesta. I contro sono che: le informazioni sono ridondanti; aggiornare il database diventa un processo più lento; raramente sono supportate dai DBMS.

viste virtuali (o viste) sono supportate da tutti i DBMS. Una interrogazione su una vista viene eseguita "ricalcolando" la vista, a meno di ottimizzazioni interne da parte del DBMS.

Le viste permettono di avere degli schemi esterni. Per cui, si mostra agli utenti solo ciò che sono interessati a conoscere. Inoltre, semplifica la scrittura di interrogazioni complesse. Diventa possibile riutilizzare programmi già esistenti su schemi ristrutturati. L'utilizzo di viste non influisce sull'efficienza delle interrogazioni, perché il DBMS ottimizza le interrogazioni.

Def. 2.13 (terza forma normale) Una relazione R con chiavi K_1, \dots, K_n è in terza forma normale se per ogni dipendenza funzionale non banale $X \rightarrow Y$, almeno una delle seguenti condizioni sono valide:

1. X è superchiave (BCNF);
2. ogni attributo Y è contenuto in almeno una tra le chiavi K_1, \dots, K_n .