

Modello relazionale

1 Proprietà delle relazioni

Una relazione è un insieme di tuple:

- all'interno di ciascuna tupla, le componenti sono ordinate (in base all'ordine in cui i domini sono specificati nel prodotto cartesiano);
- per la definizione di insieme,
 - non si ha alcun ordinamento tra le tuple;
 - le tuple di una relazione sono distinte (cioè una relazione non può contenere due o più tuple con tutte le componenti uguali).

Inoltre, una relazione è un insieme *finito*, mentre ciascun dominio può anche essere infinito.

2 Notazione posizionale

Siano:

- S una relazione di grado k ;
- t una tupla di S ;
- $i \in \{1, \dots, k\}$.

In **notazione posizionale**, $t[i]$ denota l' i -esima componente di t .

3 Attributi e notazione per nome

A ogni componente delle tuple di una relazione può essere associato un nome, detto **nome di attributo**. Allora, la coppia (nome di attributo, dominio) si dice **attributo**.

L'uso degli attributi:

- permette di denotare le componenti di ogni tupla **per nome**, piuttosto che per posizione;

- fornisce maggiori informazioni semantiche sulla proprietà rappresentata da ciascuna componente di una tupla.

4 Schema di relazione

Siano:

- S un **nome di relazione**;
- $U_S = \{A_1, A_2, \dots, A_n\}$ un insieme di nomi di attributi;
- $\text{dom} : U_S \rightarrow D$ una funzione totale che associa a ogni nome di attributo in U_S il corrispondente dominio.

La coppia $(S(A_1, A_2, \dots, A_n), \text{dom})$ è uno **schema di relazione**.

4.1 Notazione per le tuple

Una tupla t sullo schema di relazione $(S(A_1, A_2, \dots, A_n), \text{dom})$ può essere rappresentata come

$$t = [A_1 : v_1, A_2 : v_2, \dots, A_n : v_n]$$

dove v_i , con $i \in \{1, \dots, n\}$, è un valore appartenente a $\text{dom}(A_i)$, e può essere indicato anche con la notazione $t[A_i] = v_i$.

5 Schema di base di dati

Siano S_1, S_2, \dots, S_n schemi di relazioni, con nomi di relazioni distinti. L'insieme $S = \{S_1, S_2, \dots, S_n\}$ è detto **schema di base di dati**.

6 Valori nulli

A volte, non si hanno informazioni su alcune delle proprietà dell'entità rappresentata da una tupla, perché esse:

- non sono note, anche solo temporaneamente (come, per esempio, un dato opzionale che non è stato inserito);
- non sono definite per quella particolare entità (ad esempio, l'anno di laurea di uno studente non ancora laureato).

Un primo tentativo di soluzione per rappresentare l'assenza di un'informazione potrebbe essere l'uso di un valore "speciale", appartenente al dominio ma non utilizzato come dato valido (alcuni esempi potrebbero essere 0, la stringa vuota, ecc.). Non è però detto che valore esista, oppure esso potrebbe diventare un dato valido in futuro, quindi una rappresentazione del genere non è adeguata.

Per questo, è stato introdotto un valore speciale, chiamato **valore nullo** (NULL, indicato anche con un punto di domanda, ?), che:

- non fa parte di alcun dominio;
- rappresenta sia valori ignoti che valori non definiti;
- deve essere utilizzato con cautela, perché spesso ci si aspetta che alcuni attributi abbiano sempre un valore (ad esempio, la matricola di uno studente).

Nello schema di una relazione è necessario specificare quali attributi sono opzionali, cioè ammettono valori NULL, e quali sono invece obbligatori. A tale scopo, gli attributi opzionali si indicano aggiungendo un circoletto dopo il nome: ad esempio $S(A, B_o, C_o)$ indica che B e C sono opzionali, mentre A è obbligatorio.

7 Vincoli di integrità

I **vincoli di integrità (VI)** sono condizioni che il DBMS verifica per controllare che i dati siano semanticamente corretti. Tali vincoli vengono specificati in fase di definizione dello schema.

Un'istanza di una relazione si dice **corretta** se rispetta tutti i VI definiti su di essa. Il DBMS deve consentire solo istanze corrette.

Un esempio di vincolo di integrità sono gli attributi obbligatori, per i quali il DBMS controlla che non vengano inseriti valori nulli.

8 Vincoli di chiave

Siccome una relazione è un insieme, le tuple in essa contenute devono essere distinte. Tale VI è chiamato **vincolo di chiave**. Infatti, si dice **chiave** di una relazione un insieme di attributi che distingue tra loro le tuple della relazione.

Definizione formale: Sia $S(A_1, \dots, A_n)$ uno schema di relazione.¹ Un insieme $X \subseteq U_S$ è **chiave** di S se verifica entrambe le seguenti proprietà:

1. qualunque sia lo stato di S , non esistono due tuple distinte di S che abbiano lo stesso valore per tutti gli attributi in X ;

¹La funzione dom, anch'essa parte dello schema, è qui omessa per semplicità.

2. nessun sottoinsieme proprio di X verifica la proprietà 1 (in altre parole, l'insieme X non contiene attributi "superflui").

Invece, un insieme di attributi che soddisfa la proprietà 1 ma non la 2 si dice **superchiave**.

Le chiavi identificano le singole tuple di una relazione, quindi:

- garantiscono l'accessibilità a ciascun dato della base di dati;
- permettono di correlare dati appartenenti a relazioni diverse.

Per individuare le chiavi, cioè per determinare quali insiemi di attributi soddisfano le proprietà 1 e 2, è necessario analizzare il dominio applicativo. Ad esempio, in una relazione che rappresenta gli studenti di un'università, la matricola costituisce una chiave perché, per definizione, a ogni studente viene assegnata una matricola diversa; invece, nome e cognome non costituiscono una chiave perché potrebbero esserci più studenti con nomi e cognomi uguali.

In ogni caso, una relazione S qualsiasi ha sicuramente almeno una chiave: siccome, per la definizione di relazione, le tuple devono essere distinte, l'insieme U_S (tutti gli attributi di S) soddisfa sempre la proprietà 1, e, in assenza di altre chiavi, soddisfa anche la 2.

8.1 Chiavi candidate e chiave primaria

Una relazione può avere più insiemi di attributi che verificano le proprietà 1 e 2. Tutti questi insiemi si dicono **chiavi candidate**.

Bisogna poi scegliere uno di essi come **chiave primaria**, mentre gli altri prendono il nome di **chiavi alternative**.

La scelta della chiave primaria richiede particolare attenzione. In generale, si preferisce:

- la chiave candidata con il minor numero di attributi, perché i valori della chiave primaria verranno poi inseriti anche in altre tabelle per realizzare collegamenti tra le relazioni: se la chiave è grande, sarà allora necessario inserire tanti valori;
- la chiave candidata su cui ci si aspetta di fare più ricerche, visto che il DBMS userà sicuramente la chiave primaria per creare un indice.

Inoltre, negli attributi appartenenti alla chiave primaria *non sono ammessi valori nulli* (perché un valore nullo non permette di identificare una tupla), mentre questi sono consentiti nelle chiavi alternative.

8.1.1 Notazione

Nello schema di una relazione, gli attributi che costituiscono la chiave primaria vengono sottolineati.

Ad esempio, $S(\underline{A}, \underline{B}, C, \underline{D}, E)$ indica che la chiave primaria è $\{A, B, D\}$.

9 Chiavi esterne

Una **chiave esterna** modella un'associazione tra relazioni mediante una rappresentazione per valore (invece di un puntatore). Essa contiene valori corrispondenti alla chiave primaria di una tupla appartenente a un'altra relazione.

Definizione formale: Siano:

- S e S' due relazioni;
- $Y \subseteq U_{S'}$ una chiave per S' ;
- $X \subseteq U_S$ un insieme di attributi di S tale che Y e X contengano lo stesso numero di attributi e che questi ultimi abbiano domini *compatibili* (cioè non necessariamente uguali, ma che possono essere confrontati, come ad esempio un intero e un contatore);

X è una **chiave esterna** di S su S' se, qualsiasi siano gli stati di S ed S' , per ogni tupla t di S esiste una tupla t' di S' tale che $t[X] = t'[Y]$.

Allora:

- S viene detta **relazione referente**;
- S' viene detta **relazione riferita**.

9.1 Notazione

Nello schema della relazione referente, gli attributi che costituiscono una chiave esterna vengono indicati scrivendo in apice il nome della relazione riferita.

Ad esempio,

$$\begin{array}{l} \text{Docente}(\underline{\text{Matricola}}, \dots) \\ \text{Corso}(\underline{\text{Codice}}, \dots, \text{MatrDocente}^{\text{Docente}}, \dots) \end{array}$$

significa che la relazione Corso ha una chiave esterna riferita a Docente e composta dall'attributo MatrDocente.

9.2 Osservazioni

- I nomi degli attributi della chiave (primaria) nella relazione riferita e della chiave esterna nella relazione referente *non devono necessariamente* essere uguali (ma, se lo sono, semplificano alcune operazioni).
- Gli attributi hanno sicuramente nomi diversi quando la relazione referente e quella riferita coincidono, cioè la chiave esterna fa riferimento a tuple della relazione stessa. Ad esempio, una relazione Persona potrebbe contenere due chiavi esterne per rappresentare il padre e la madre:

$$\text{Persona}(\underline{\text{CF}}, \dots, \text{Padre}^{\text{Persona}}, \text{Madre}^{\text{Persona}}, \dots)$$

- Una relazione può avere più chiavi esterne, eventualmente anche sulla stessa relazione.
- Le chiavi esterne devono essere specificate esplicitamente nello schema di relazione: il fatto di avere attributi con gli stessi nomi e con domini compatibili non dà garanzie sull'integrità referenziale.
- Uno stesso valore della chiave esterna può essere ripetuto in più tuple della relazione referente. Allora, tutte queste tuple saranno correlate alla stessa tupla della relazione riferita. Inoltre, non è detto che le chiavi esterne facciano riferimento a tutte le tuple della relazione riferita.
- Le chiavi esterne possono assumere valori nulli (a meno che ciò non venga impedito mediante un apposito vincolo). Una tupla con una chiave esterna nulla non ha un collegamento verso la relazione riferita.

10 Integrità referenziale

Il vincolo di integrità imposto dalla definizione di chiave esterna (cioè che i valori assunti da una chiave esterna nella relazione referente devono esistere come valori della chiave primaria nella relazione riferita) è chiamato **vincolo di integrità referenziale**.

In pratica, verificando che tale vincolo sia soddisfatto, il DBMS garantisce che le associazioni rappresentate dalle chiavi esterne siano coerenti.

10.1 Violazioni dell'integrità referenziale

L'integrità referenziale può essere violata da:

- inserimenti e modifiche (del valore della chiave esterna) nella relazione referente;
- cancellazioni e modifiche (del valore della chiave primaria) nella relazione riferita.

10.1.1 Operazioni sulla relazione referente

Quando vengono inserite o modificate delle tuple nella relazione referente, il DBMS controlla che i nuovi valori della chiave esterna esistano come valori della chiave primaria nella relazione riferita. In caso contrario, l'operazione di inserimento/modifica viene rifiutata per mantenere l'integrità referenziale.

10.1.2 Operazioni sulla relazione riferita

Se si cancellassero delle tuple nella relazione riferita, oppure se si modificassero alcuni valori della chiave primaria, le chiavi esterne nella relazione referente potrebbero contenere ancora i valori vecchi/cancellati della chiave primaria, violando così l'integrità referenziale.

Il DBMS mette a disposizione varie strategie per gestire questa situazione:

- rifiutare l'operazione;
- cancellare/modificare *a cascata* anche le tuple corrispondenti nella relazione referente;
- sostituire i valori delle chiavi esterne che diventerebbero non validi con valori nulli o di default.

Quale sia la più adatta tra queste soluzioni dipende dal dominio applicativo.