

Il modello relazionale

I database relazionali sono il tipo di database attualmente piu' diffuso. I motivi di questo successo sono fondamentalmente due:

1. forniscono sistemi semplici ed efficienti per rappresentare e manipolare i dati
2. si basano su un modello, quello relazionale, con solide basi teoriche

Il modello relazionale e' stato proposto originariamente da E.F. Codd in un ormai famoso articolo del 1970. Grazie alla sua coerenza ed usabilita', il modello e' diventato negli anni '80 quello piu' utilizzato per la produzione di DBMS.

La struttura fondamentale del modello relazionale e' appunto la "relazione", cioe' una tabella bidimensionale costituita da righe (tuple) e colonne (attributi). Le relazioni rappresentano le entita' che si ritiene essere interessanti nel database. Ogni istanza dell'entita' trovera' posto in una tupla della relazione, mentre gli attributi della relazione rappresenteranno le proprieta' dell'entita'. Ad esempio, se nel database si dovranno rappresentare delle persone, si potra' definire una relazione chiamata "Persone", i cui attributi descrivono le caratteristiche delle persone (Figura 2). Ciascuna tupla della relazione "Persone" rappresentera' una particolare persona.

In realta', volendo essere rigorosi, una relazione e' solo la definizione della struttura della tabella, cioe' il suo nome e l'elenco degli attributi che la compongono. Quando essa viene popolata con delle tuple, si parla di "istanza di relazione". Percio' la precedente Figura 2 rappresenta un'istanza della relazione persona. Una rappresentazione della definizione di tale relazione potrebbe essere la seguente:

Persone (nome, cognome, data_nascita, sesso, stato_civile)

Nel seguito si indicheranno entrambe (relazione ed istanza di relazione) con il termine "relazione", a meno che non sia chiaro dal contesto a quale accezione ci si riferisce.

Le tuple in una relazione sono un insieme nel senso matematico del termine, cioe' una collezione non ordinata di elementi differenti. Per distinguere una tupla da un'altra si ricorre al concetto di "chiave primaria", cioe' ad un insieme di attributi che permettono di identificare univocamente una tupla in una relazione. Naturalmente in una relazione possono esserci piu' combinazioni di attributi che permettono di identificare univocamente una tupla ("chiavi candidate"), ma fra queste ne verra' scelta una sola da utilizzare come chiave primaria. Gli attributi della chiave primaria non possono assumere il valore null (che significa un valore non determinato), in quanto non permetterebbero piu' di identificare una particolare

tupla in una relazione. Questa proprietà delle relazioni e delle loro chiavi primarie va sotto il nome di integrità delle entità (entity integrity).

Spesso per ottenere una chiave primaria "economica", cioè composta da pochi attributi facilmente manipolabili, si introducono uno o più attributi fittizi, che conterranno dei codici identificativi univoci per ogni tupla della relazione.

Ogni attributo di una relazione è caratterizzato da un nome e da un dominio. Il dominio indica quali valori possono essere assunti da una colonna della relazione. Spesso un dominio viene definito attraverso la dichiarazione di un tipo per l'attributo (ad esempio dicendo che è una stringa di dieci caratteri), ma è anche possibile definire domini più complessi e precisi. Ad esempio per l'attributo "sesso" della nostra relazione "Persone" possiamo definire un dominio per cui gli unici valori validi sono 'M' e 'F'; oppure per l'attributo "data_nascita" potremmo definire un dominio per cui vengono considerate valide solo le date di nascita dopo il primo gennaio del 1960, se nel nostro database non è previsto che ci siano persone con data di nascita antecedente a quella. Il DBMS si occuperà di controllare che negli attributi delle relazioni vengano inseriti solo i valori permessi dai loro domini. Caratteristica fondamentale dei domini di un database relazionale è che siano "atomici", cioè che i valori contenuti nelle colonne non possano essere separati in valori di domini più semplici. Più formalmente si dice che non è possibile avere attributi multivalore (multivalued). Ad esempio, se una caratteristica delle persone nel nostro database fosse anche quella di avere uno o più figli, non sarebbe possibile scrivere la relazione Persone nel seguente modo:

Persone (nome, cognome, data_nascita, sesso, stato_civile, figli)

Infatti l'attributo figli è un attributo non-atomico, sia perché una persona può avere più di un figlio, sia perché ogni figlio avrà varie caratteristiche che lo descrivono. Per rappresentare queste entità in un database relazionale bisogna definire due relazioni:

Persone(*numero_persona, nome, cognome, data_nascita, sesso, stato_civile)

Figli(*numero_persona, *nome_cognome, eta, sesso)

Nelle precedenti relazioni gli asterischi (*) indicano gli attributi che compongono le loro chiavi primarie. Si noti l'introduzione nella relazione Persone dell'attributo numero_persona, attraverso il quale si assegna a ciascuna persona un identificativo numerico univoco che viene utilizzato come chiave primaria. Queste relazioni contengono solo attributi atomici. Se una persona ha più di un figlio, essi saranno rappresentati in tuple differenti della relazione Figli. Le varie caratteristiche dei figli sono rappresentate dagli attributi della relazione Figli. Il legame fra le due relazioni è costituito dagli attributi numero_persona che compaiono in entrambe le relazioni e che permettono di assegnare ciascuna tupla della relazione figli ad una particolare tupla della relazione Persone. Più formalmente si dice che l'attributo numero_persona della relazione Figli è una chiave esterna (foreign key) verso la relazione Persone. Una chiave esterna è una combinazione di

attributi di una relazione che sono chiave primaria per un'altra relazione. Una caratteristica fondamentale dei valori presenti in una chiave esterna e' che, a meno che non siano null, devono corrispondere a valori esistenti nella chiave primaria della relazione a cui si riferiscono. Nel nostro esempio cio' significa che non puo' esistere nella relazione Figli una tupla con un valore dell'attributo numero_persona, senza che anche nella relazione Persone esista una tupla con lo stesso valore per la sua chiave primaria. Questa proprieta' va sotto il nome di integrita' referenziale (referential integrity)

Uno dei grandi vantaggi del modello relazionale e' che esso definisce anche un'algebra, chiamata appunto "algebra relazionale". Tutte le manipolazioni possibili sulle relazioni sono ottenibili grazie alla combinazione di cinque soli operatori: REStRICT, PROJECT, TIMES, UNION e MINUS. Per comodita' sono stati anche definiti tre operatori addizionali che comunque possono essere ottenuti applicando i soli cinque operatori fondamentali: JOIN, INTERSECT e DIVIDE. Gli operatori relazionali ricevono come argomento una relazione o un insieme di relazioni e restituiscono una singola relazione come risultato.

Vediamo brevemente questi otto operatori:

REStRICT: restituisce una relazione contenente un sottoinsieme delle tuple della relazione a cui viene applicato. Gli attributi rimangono gli stessi.

PROJECT: restituisce una relazione con un sottoinsieme degli attributi della relazione a cui viene applicato. Le tuple della relazione risultato vengono composte dalle tuple della relazione originale in modo che continuino ad essere un insieme in senso matematico.

TIME: viene applicato a due relazioni ed effettua il prodotto cartesiano delle tuple. Ogni tupla della prima relazione viene concatenata con ogni tupla della seconda.

JOIN: vengono concatenate le tuple di due relazioni in base al valore di un insieme dei loro attributi.

UNION: applicando questo operatore a due relazioni compatibili, se ne ottiene una contenente le tuple di entrambe le relazioni. Due relazioni sono compatibili se hanno lo stesso numero di attributi e gli attributi corrispondenti nelle due relazioni hanno lo stesso dominio.

MINUS: applicato a due relazioni compatibili, ne restituisce una terza contenente le tuple che si trovano solo nella prima relazione.

INTERSECT: applicato a due relazioni compatibili, restituisce una relazione contenente le tuple che esistono in entrambe le relazioni.

DIVIDE: applicato a due relazioni che abbiano degli attributi comuni, ne restituisce una terza contenente tutte le tuple della prima relazione che possono essere fatte corrispondere a tutti i valori della seconda relazione.

Nelle seguenti tabelle, a titolo di esempio, sono raffigurati i risultati dell'applicazione di alcuni operatori relazionali alle relazioni Persone e Figli. Come nomi per le relazioni risultato si sono utilizzate le espressioni che le producono.

Person	numero_persona	nome	cognome	data_nascita	sex	stato_civile
2	Mario	Rossi	29/03/1965	M	Coniugato	
1	Giuseppe	Russo	15/11/1972	M	Celibe	
3	Alessandra	Mondella	13/06/1970	F	Nubile	

Figli	numero_persona	nome_cognome	eta	sex
2	Maria	Rossi	3	F
2	Gianni	Rossi	5	M

RESTRiCT (Persone)

sex='M'	numero_persona	nome	cognome	data_nascita	sex	stato_civile
2	Mario	Rossi	29/03/1965	M	Coniugato	
1	Giuseppe	Russo	15/11/1972	M	Celibe	

PROJECT sesso (Persone)sesso

M

F

JOIN (Persone, Figli)

Persone(numero_persona)=Figli(numero_persona)n.	nome	cognome	nascita	sezzo	stato_civile	nome	eta'	sezzo
2	Mario	Rossi	29/03/1965	M	Coniugato	Maria Rossi	3	F
2	Mario	Rossi	29/03/1965	M	Coniugato	Gianni Rossi	5	M

I database relazionali compiono tutte le operazioni sulle tabelle utilizzando l'algebra relazionale, anche se normalmente non permettono all'utente di utilizzarla. L'utente interagisce con il database attraverso un'interfaccia differente, il linguaggio SQL, un linguaggio dichiarativo che permette di descrivere insiemi di dati. Le istruzioni SQL vengono scomposte dal DBMS in una serie di operazioni relazionali.