

Basi di dati e Web 2008/2009

Lezione del 23 febbraio 2009 - Appunti

Progettazione logica

La progettazione logica di una base di dati è un'attività che prevede come punto di partenza il documento contenente lo schema concettuale. Tuttavia, per comprendere i criteri e le tecniche di rappresentazione dell'informazione nel modello relazionale, supponiamo di non avere lo schema concettuale e di voler progettare direttamente lo schema logico di una base di dati.

Chi si accosta in modo intuitivo a questa attività potrebbe porsi la seguente domanda: PERCHE' SCARTARE LA SOLUZIONE MONOTABELLARE (vale a dire quella dove rappresento tutti gli attributi in un'unica tabella)?

Vediamo un esempio:

TABELLA Proprietari-Appartamenti

Codice fiscale	Cognome	Nome	Data di nascita	Codice ecografico	Via	N° civico	Subalterno	Categoria catastale
RSSRBT	Rossi	Roberto	10/10/41	AB508	Roma	10	A	A4
RSSRBT	Rossi	Roberto	10/10/41	AC101	Garibaldi	5	B	B2
BNCMRA	Bianchi	Maria	1/12/54	AB508	Roma	10	A	A4

Si noti che in questo esempio si è voluto rappresentare con una sola tabella le informazioni che descrivono i proprietari di appartamenti residenti in un comune e i loro diritti di proprietà. Si noti inoltre che, a causa della rappresentazione monotabellare, i valori di alcuni attributi sono stati replicati generando RIDONDANZA (valori nei rettangoli rossi).

La ridondanza crea problemi in una base di dati per i seguenti motivi:

- Genera spreco di memoria (anche se questo è un inconveniente poco importante, vista la disponibilità di memoria negli attuali dispositivi di memorizzazione di massa)
- Genera ANOMALIE (questo è il vero inconveniente):
 - ANOMALIA DI AGGIORNAMENTO: si verifica quando per aggiornare il valore di un attributo è necessario aggiornare valori su tuple diverse. Questo può generare inconsistenze se non vengono aggiornate tutte le repliche del valore dell'attributo. Ad esempio se per un errore materiale devo modificare il numero civico dell'appartamento sito in via Roma, dovrò aggiornare due tuple e se ne aggiornano solo una la base di dati si troverà in uno stato inconsistente.
 - ANOMALIA DI INSERIMENTO: si verifica quando per inserire un'istanza di informazione è necessario valorizzare attributi della tupla che rappresentano dati non noti al momento dell'inserimento (oppure è necessario inserire valori nulli sugli attributi non noti).
 - ANOMALIA DI CANCELLAZIONE: si verifica quando per cancellare un'istanza di informazione è necessario cancellare anche valori di attributi che dovrebbero restare nella base di dati (oppure è necessario inserire valori nulli negli attributi dell'istanza da cancellare).

Per evitare le ANOMALIE sopra elencate è necessario suddividere la tabella unica in più tabelle (decomposizione). La suddivisione in più tabelle non può essere fatta in modo qualsiasi; ad esempio applicare un puro partizionamento non è una strategia corretta. Considerando l'esempio precedente, la seguente partizione non sarebbe corretta:

TABELLA Proprietari

Codice fiscale	Cognome	Nome	Data di nascita
RSSRBT	Rossi	Roberto	10/10/41
BNCMRA	Bianchi	Maria	1/12/54

TABELLA Appartamenti

Codice ecografico	Via	N° civico	Subalterno	Categoria catastale
AB508	Roma	10	A	A4
AC101	Garibaldi	5	B	B2

Infatti, in questa situazione si è perso il legame logico tra proprietari ed appartamenti. Come è possibile rimediare? Esiste un modo per rappresentare legami logici tra tuple nel modello relazionale?

Si noti che nel modello relazionale non esistono puntatori o riferimenti a tuple o qualsiasi meccanismo simile (tipico degli approcci "object-oriented"). L'unico modo di rappresentare informazione è quello di rappresentare tuple di valori. Quindi per rappresentare il legame logico tra proprietari e appartamenti occorre memorizzare tuple che associano ogni proprietario ai suoi appartamenti. Ora, per non ricostruire esattamente la tabella unica proposta inizialmente, esiste un approccio che consiste nello scegliere tra gli attributi di una tabella l'attributo o gli attributi "rappresentanti": tali attributi devono avere la proprietà di identificare una tupla della tabella (nelle prossime lezioni introdurremo il concetto di chiave primaria di una tabella per indicare esattamente questo/i attributo/i).

A questo punto per rappresentare il legame logico tra tuple basta memorizzare in una tabella aggiuntiva le coppie di tuple che rappresentano istanze di tale legame. Quindi per l'esempio sopra esposto va aggiunta una tabella come segue:

TABELLA Proprietà

Proprietario	Appartamento
RSSRBT	AB508
BNCMRA	AB508
RSSRBT	AC101

Ogni riga di tale tabella rappresenta un legame logico tra tuple.

E' evidente che si è introdotta della ridondanza, nel senso che gli attributi "Codice fiscale" e "Codice ecografico" sono stati replicati anche nella tabella Proprietà; tuttavia tale replicazione è funzionale a rappresentare un legame logico che altrimenti non sarebbe rappresentato. Per tale motivo si tratta di ridondanza "utile" e al fine di assicurarne la consistenza i sistemi relazionali prevedono meccanismi per mantenere allineati i valori replicati ("chiavi esportate") rispetto all'attributo di riferimento ("chiave primaria"). Nel nostro caso le chiavi primarie sono costituite: dall'attributo "Codice fiscale" per la tabella Proprietari e dall'attributo "Codice ecografico" per la tabella Appartamenti. Le chiavi esportate sono costituite dall'attributo Proprietario della tabella Proprietà, che esporta la chiave di Proprietari, e dall'attributo Appartamento di Proprietà, che esporta la chiave di Proprietari.

N.B.: i legami tra tuple nel modello relazionale si rappresentano attraverso l'esportazione degli attributi "rappresentanti" (o chiave primaria) di una tabella dentro un'altra tabella e il legame si intende stabilito tra due tuple quando esse presentano gli stessi valori negli attributi comuni.

Il modello relazionale è VALUE-BASED in quanto ogni informazione viene rappresentata attraverso tuple di valori (non ci sono meccanismi simili ai puntatori o ai riferimenti ad oggetti). Ciò implica che:

- È completamente indipendente dalla rappresentazione fisica (non ci sono puntatori o riferimenti a tuple).
- È facile trasferire dati da una base di dati relazionale all'altra in quanto tutto è rappresentato in valori significativi e visibili all'utente.
- Viene rappresentato solo ciò che è rilevante per il sistema informativo.

Informazione incompleta e valori nulli (NULL)

Dalla definizione di relazione deriva che ogni tupla deve contenere valori appartenenti ai domini su cui la relazione è definita.

$$R(X) \quad X=\{A_1: D_1, \dots, A_n: D_n\}$$
$$t \in r \quad t[A_i] \in D_i, \dots, t[A_n] \in D_n$$

Tuttavia, non sempre in una base di dati reale esistono i valori per tutti gli attributi di una tupla. In particolare si possono presentare i seguenti casi:

- Il valore dell'attributo è **SCONOSCIUTO**: esiste nella realtà ma in questo momento non è noto alla base di dati
- Il valore dell'attributo è **INESISTENTE**: non esiste per questa tupla un valore per l'attributo e non ne esisterà mai uno.
- Il valore dell'attributo è **SCONOSCIUTO** o **INESISTENTE**.

Per poter trattare queste situazioni in una base di dati relazionale si introduce un valore speciale nel modello relazionale detto **VALORE NULLO (NULL VALUE)**. Quindi, gli attributi di una tupla possono assumere o un valore del dominio ad essi associato oppure il valore nullo.

Si noti infine che la presenza di valori nulli è accettabile solo in alcuni attributi; negli attributi che rappresentano "chiavi" o "chiavi esportate" la presenza di valori nulli può rendere inutilizzabile l'informazione.