A2 - 3.Progettazione e normalizzazione di un DB relaz. - decomposizione lossless

pag. A47 - A55 (parte sesta)

decomposizione lossless

Riprendiamo in esame l'esempio dell'allocazione delle sale operatorie di un ospedale:

Interventi (Paziente, DataIntervento, OraIntervento, Chirurgo, Sala);

Abbiamo visto che per ridurre la ridondanza e normalizzare lo schema originale, è necessario effettuare una decomposizione dello schema stesso in due più piccoli:

OccupazioneSale (Chirurgo, DataIntervento, Sala);

Interventi (Paziente, DataIntervento, OraIntervento, Chirurgo);

Chieurgo	DataInterevnto	Sala
De Bakey	25/10/2005	Sala1
Romano	25/10/2005	Sala2
Veronesi	26/10/2005	Sala1

OccupazioneSale

Paziente	DataInterevnto	OraIntervento	Chrurgo
Bianchi	25/10/2005	8.00	De Bakey
Rossi	25/10/2005	8.00	Romano
Negri	26/10/2005	9.30	Veronesi
Viola	25/10/2005	10.30	De Bakey
Verdi	25/10/2005	11.30	Romano

Interventi

decomposizione lossless

Se però volessimo sapere in quale sala operatoria verrà operato il paziente *Negri*, non saremmo in grado di ottenere facilemente questa informazione, perché i nomi dei pazienti e la lista delle sale operatorie si trovano su due tabelle differenti.

Sarà quindi necessario rimettere insieme i dati delle due tabelle per ricostruire la tabella originale, prima di poter effettuare tale operazione.

Il paziente Negri verrà operato nella sala operatoria n.1

Chieurgo	DataInterevnto
De Bakey	25/10/2005
Romano	25/10/2005
Veronesi	26/10/2005

OccupazioneSale

Paziente	DataInterevnto	OraIntervento	Chrurgo	Sala
Bianchi	25/10/2005	8.00	De Bakey	Sala1
Rossi	25/10/2005	8.00	Romano	Sala2
Negri	26/10/2005	9.30	Veronesi	Sala1
Viola	25/10/2005	10.30	De Bakey	Sala1
Verdi	25/10/2005	11.30	Romano	Sala2

Interventi

decomposizione lossless

Questa operazione può essere effettuata usando un operatore dell'algebra relazionale detto di ***congiunzione*** (*natural join*) che **correla dati da tabelle diverse, sulla base di valori uguali in campi con lo stesso nome**.

Paziente	DataInterevnto	OraIntervento	Chrurgo
Bianchi	25/10/2005	8.00	De Bakey
Rossi	25/10/2005	8.00	Romano
Negri	26/10/2005	9.30	Veronesi
Viola	25/10/2005	10.30	De Bakey
Verdi	25/10/2005	11.30	Romano

Interventi

Chirurgo	DataInterevnto	Sala
De Bakey	25/10/2005	Sala1
Romano	25/10/2005	Sala2
Veronesi	26/10/2005	Sala1

OccupazioneSale

Il *join naturale* mette insieme le righe delle due tabelle la dove coincide il valore della coppia di campi in comune: *Chirurgo* e *DataIntervento*.

Paziente	DataInterevnto	OraIntervento	Chrurgo	Sala
Bianchi	25/10/2005	8.00	De Bakey	Sala1
Rossi	25/10/2005	8.00	Romano	Sala2
Negri	26/10/2005	9.30	Veronesi	Sala1
Viola	25/10/2005	10.30	De Bakey	Sala1
Verdi	25/10/2005	11.30	Romano	Sala2

Interventi

decomposizione lossless

A prima vista risulta strano il fatto di applicare un procedimento di decomposizione degli schemi di relazione per normalizzarli, quando in fase di ricerca dei dati è spesso necessario utilizzare operatori che applichino il processo inverso di ricomposizione.

La decomposizione ha come scopo la limitazione della ridondanza dei dati al fine di garantire una corretta esecuzione delle operazioni di modifica dei dati di una tabella (inserimento, aggiornamento, cancellazione).

Il prezzo da pagare per garantire l'integrità dei dati è quello di dover ricomporre le tabelle mediante specifici operatori nel momento in cui si effettuano operazioni di ricerca dei dati.

decomposizione lossless

Il processo di normalizzazione tramite decomposizione deve quindi essere tale da rendere possibile la ricomposizione delle tabelle una volta che queste siano state decomposte in schemi più piccoli che non presentano anomalie.

Per questo motivo, ogni decomposizione che effettuiamo in uno schema di relazione deve garantire le seguenti proprietà:

- ***join senza perdita*** (*lossless join*) che assicura che negli schemi di relazione creati dopo una decomposizione non si presenti il problema di generazione di *tuple spurie*; questa proprietà viene indicata come **decomposizione senza perdita**;
- *conservazione delle dipendenze* che garantisce che ogni dipendenza funzionale sia rappresentata in qualcuno degli schemi risultanti dopo la decomposizione.

decomposizione lossless

Quello della decomposizione senza perdita è un requisito irrinunciabile nel processo di normalizzazione di uno schema di relazione; riprendiamo in esame l'esempio seguente:

Insegnamenti (docente, materia, studente);

Come visto in precedenza, risulta in 3NF, ma non in BCNF a causa della dipendenza funzionale:

docente → materia

dal momento che *docente* non è superchiave per la tabella *Insegnamenti*.

La decomposizione per renderla in BCNF non è però così immediata perché possono esserci più alternative.

docente	materia	studente
Turing	Informatica	Rossi
Codd	Informatica	Neri
Madnick	Sistemi	Neri
Donovan	Elettronica	Neri
Turing	Informatica	Bianchi
Knuth	Sistemi	Bianchi
Wirth	Informatica	Verdi
Codd	Informatica	Grigi
Madnick	Sistemi	Rossi

decomposizione lossless

Potremmo decomporre lo schema originale in due schemi che comprendono rispettivamente i campi:

1° soluzione: (*docente*, *studente*) e (*studente*, *materia*):

docente	studente
Turing	Rossi
Codd	Neri
Madnick	Neri
Donovan	Neri
Turing	Bianchi
Knuth	Bianchi
Wirth	Verdi
Codd	Grigi
Madnick	Rossi

studente	materia
Rossi	Informatica
Neri	Informatica
Neri	Sistemi
Neri	Elettronica
Bianchi	Informatica
Bianchi	Sistemi
Verdi	Informatica
Grigi	Informatica
Rossi	Sistemi

decomposizione lossless

Potremmo decomporre lo schema originale in due schemi che comprendono rispettivamente i campi:

2° soluzione: (*docente*, *materia*) e (*studente*, *materia*):

docente	materia
Turing	Informatica
Codd	Informatica
Madnick	Sistemi
Donovan	Elettronica
Knuth	Sistemi
Wirth	Informatica

studente	materia
Rossi	Informatica
Neri	Informatica
Neri	Sistemi
Neri	Elettronica
Bianchi	Informatica
Bianchi	Sistemi
Verdi	Informatica
Grigi	Informatica
Rossi	Sistemi

decomposizione lossless

Potremmo decomporre lo schema originale in due schemi che comprendono rispettivamente i campi:

3° soluzione: (*docente*, *materia*) e (*docente*, *studente*):

docente	materia
Turing	Informatica
Codd	Informatica
Madnick	Sistemi
Donovan	Elettronica
Knuth	Sistemi
Wirth	Informatica

docente	studente
Turing	Rossi
Codd	Neri
Madnick	Neri
Donovan	Neri
Turing	Bianchi
Knuth	Bianchi
Wirth	Verdi
Codd	Grigi
Madnick	Rossi

decomposizione lossless

In tutti e tre i casi si perde la dipendenza funzionale:

(studente, materia) → docente

e non potrebbe essere altrimenti, comunque, l’unica scomposizione corretta è la terza perché effettuando un’operazione di congiunzione delle tabelle, non vengono generate righe spurie, ma viene ricostruita correttamente la tabella originale.

decomposizione lossless

Nel primo caso, abbiamo decomposto lo schema di relazione originale nei due schemi:

(docente, studente) e *(studente, materia)*;

Applicando l'operatore di *congiunzione* alle tabelle corrispondenti, sulla base dei valori uguali del campo *studente*, otteniamo la tabella mostrata di lato, nella quale sono state evidenziate le righe spurie, cioè quelle inesistenti nella tabella originale.

Il *join naturale* mette insieme le righe delle due tabelle la dove coincide il valore dell'unico campo in comune, *studente*.

docente	materia	studente
Madnick	Informatica	Rossi
Turing	Informatica	Rossi
Codd	Informatica	Neri
Donovan	Informatica	Neri
Madnick	Informatica	Neri
Codd	Sistemi	Neri
Donovan	Sistemi	Neri
Madnick	Sistemi	Neri
Codd	Elettronica	Neri
Donovan	Elettronica	Neri
Madnick	Elettronica	Neri
Knuth	Informatica	Bianchi
Turing	Informatica	Bianchi
Knuth	Sistemi	Bianchi
Turing	Sistemi	Bianchi
Wirth	Informatica	Verdi
Codd	Informatica	Grigi
Madnick	Sistemi	Rossi
Turing	Sistemi	Rossi

decomposizione lossless

La seconda decomposizione mostrata, genera gli schemi seguenti:

(docente, materia) e *(studente, materia)*;

la composizione avviene sulla base del valore dell'attributo *materia*, e produce la tabella mostrata al lato.

Anche in questo caso vengono prodotte diverse righe spurie.

docente	materia	studente
Turing	Informatica	Grigi
Turing	Informatica	Verdi
Turing	Informatica	Bianchi
Turing	Informatica	Neri
Turing	Informatica	Rossi
Codd	Informatica	Grigi
Codd	Informatica	Verdi
Codd	Informatica	Bianchi
Codd	Informatica	Neri
Codd	Informatica	Rossi
Madnick	Sistemi	Rossi
Madnick	Sistemi	Bianchi
Madnick	Sistemi	Neri
Donovan	Elettronica	Neri
Knuth	Sistemi	Rossi
Knuth	Sistemi	Bianchi
Knuth	Sistemi	Neri
Wirth	Informatica	Grigi
Wirth	Informatica	Verdi
Wirth	Informatica	Bianchi
Wirth	Informatica	Neri
Wirth	Informatica	Rossi

decomposizione lossless

L’unica scomposizione corretta è la terza:

(docente, materia) e *(docente, studente)*;

perché applicando il join naturale sulla base del campo in comune *docente*, non sono generate righe spurie, ma viene ricostruita correttamente la tabella originale.

docente	materia	studente
Turing	Informatica	Rossi
Codd	Informatica	Neri
Madnick	Sistemi	Neri
Donovan	Elettronica	Neri
Turing	Informatica	Bianchi
Knuth	Sistemi	Bianchi
Wirth	Informatica	Verdi
Codd	Informatica	Grigi
Madnick	Sistemi	Rossi

decomposizione lossless

Teorema:

Sia data una tabella T(X), con X insieme degli attributi di T, e due sottoinsiemi A e B di X tali che $A \cup B = X$; siano date, inoltre, due tabelle T_1 e T_2 rispettivamente con insiemi di attributi A e B. Condizione sufficiente affinché la decomposizione di X su A e B sia «senza perdita» è che l'insieme $C = A \cap B$ sia una superchiave per $T_1(A)$ o $T_2(B)$.