Appunti Basi di Dati

Luca Seggiani

21 Marzo 2024

1 Relazioni Derivate

Una **relazione di base** è una relazione il cui contenuto è autonomo. Una **relazione derivata** è una relazione il cui contenuto è funzione di altre relazioni. Si possono cosi avere rappresentazioni diverse per gli stessi dati, definite mediante interrogazioni, che possono rivolgersi a relazioni di base come ad altre relazioni derivate. Esistono due tipi di relazioni derivate:

- **Viste materializzate**, dove i risultati sono precalcolati;
- **Viste virtuali**, o più semplicemente viste. Sono le più comuni.

Viste materializzate
Sono relazioni derivate memorizzate nella base di dati. Il loro vantaggio è che sono precalcolate, e quindi immediatamente disponibili. I loro svantaggi sono che:

- Sono ridondanti;
- Appesantiscono gli aggiornamenti;
- Sono raramente supportate dai DBMS.

Viste virtuali
Relazioni derivate non memorizzate nella base di dati, e quindi ricalcolate ad ogni accesso. Sono supportate da tutti i DBMS.

Interrogazioni su viste
Le interrogazioni su viste sono eseguite sostituendo alla vista la sua definizione. Le viste possono semplificare la scrittura di interrogazioni, espressioni complesse, e sotto-espressioni ripetute. L'uso delle viste virtuali inoltre non

influisce sull'efficienza delle interrogazioni. Ad esempio, supponiamo di avere le seguenti relazioni:

$$R_1(ABC), R_2(DEF), R_3(GH)$$

e di definire una vista R:

$$R = \sigma_{A>D}(R_1 \bowtie R_2)$$

Un'interrogazione potrà a questo punto essere definita:

- Senza vista:
$$\sigma_{B=G}(\sigma_{A>D}(R_1 \bowtie R_2) \bowtie R_3)$$

  item Con vista:
$$\sigma_{B=G}(R \bowtie R_3)$$

Viste e aggiornamenti
Aggiornare una vista significa modificare le relazioni di base in modo che la vista ricalcolata rispecchi l'aggiornamento. L'aggiornamento sulle relazioni di base dovrebbe essere associato univocamente a quello specificato sulla vista, ma questo non è sempre vero. Per questo motivo ben pochi aggiornamenti sono ammissibili sulle viste.

2 Calcolo relazionale

Il calcolo relazionale è una famiglia di linguaggi dichiaratvi basati sul calcolo dei predicati del primo ordine. Ne esistono diverse versioni:

- Calcolo relazionale sui domini, in breve calcolo dei domini;
- Calcolo su n-uple con dichiarazione di range, in breve calcolo delle tuple, usato in SQL.

Assunzioni
I simboli di predicato corrispondono alle relazioni presenti nelle basi di dati, più alcuni predicati standard come l'uguaglianza e la disuguaglianza. Non compaiono simboli di funzione. Nel calcolo relazionale vengono utilizzate prevalentemente formule aperte, cioè formule con variabili libere, il cui valore di verità dipende dai valori assegnati alle variabili libere. Il risultato di un'interrogazione (formula aperta) è costituito da tutte le tuple di valori che sostituiti

alle variabili libere, la rendono vera. In coerenza con quanto fatto in algebra relazionale, useremo una notazione non posizionale (attributi con nome, etichettati).

Calcolo dei domini

Vediamo innanzitutto la sintassi: nel calcolo dei domini un'espressone ha forma:

$$\{A_1 : x_1, ..., A_k : x_k \mid f\}$$

dove:

- $A_1, ..., A_k$ sono attributi distinti (che possono anche non comparire nello schema della base di dati).
- $x_1, ..., x_K$ sono variabili (che assumiamo essere distinte, non è necessario).
- $A_1 : x_1, ..., A_k : x_k$ è la *target list* (lista degli obiettivi) e descrive il risultato.
- f è una formula costruita a partire da formule atomiche utilizzando eventualmente i connettivi booleani e i quantificatori logici $\forall x$, $\exists x$, con x variabile (logica del primo ordine).

Il risultato di un'espressione nel calcolo dei domini è una relazione su $A_1, ..., A_k$, contenente n-uple in $x_1, ..., x_2$ che rendono vera la formula f sull'istanza di basi di dati considerata. F rispetto all'istanza

Formule atomiche del calcolo dei domini

Il predicato f è costituito da formule atomiche. Una formula atomica è definita come una delle tre forme:

- **Schemi di relazione**

  $$R(A_1 : x_1..., A_p : x_p)$$

  dove $R(A_1, ..., A_P)$ è uno schema di relazione.

- **Operatori fra variabili**

  $$x_i \; OP \; x_j$$

  dove x e y sono variabili e OP è un operatore di confronto (uguale, non uguale, maggiore, minore e varianti strette).

- **Operatori fra variabili e costanti**

  $$x_i \; OP \; c, \quad c \; OP \; x_i$$

  dove c è una costante nel dominio A_i di x_i.

Definiamo allora alcune proprietà delle formule:

- Le formule atomiche sono formule.
- Se f è una formula, allora anche $\neg f$ è una formula.
- Se f_1 e f_2 sono formule, allora anche $f_1 \wedge f_2$ e $f_1 \vee f_2$ sono formule.
- Se f è una formula e x una variabile, allora anche $\exists x(f)$ e $\forall x(f)$ sono formule, dove $\exists$ and $\forall$ sono qualificatori (rispettivamente esistenziale e universale).
- Per convenienza, si raggruppano le variabili usate da quantificatori simili: ad esempio, $\exists x(\exists y(f))$ si può scrivere come $\exists x, y(f)$.

Valore di verità
L'inclusione di una data n-upla o meno da un'espressione è determinato dal valore di verità della sua formula. Il valore di verità di una formula è così definito (sulle formule atomiche):

- Una formula atomica $R(A_1 : x_1, ..., A_p : x_p)$ è vera sui valori $x_1, ..., x_p$ che costituiscono una n-upla valida di R.
- Una formula atomica $x\ \theta\ y$ o $x\ \theta\ c$ (con c costante) è vera sui valori x, y che soddisfano la condizione θ.
- Il valore di verità degli operatori logici (congiunzione $\wedge$, disgiunzione $\vee$, negazione $\not)$è definito nel modo usuale.
- Una formula della forma $\exists x(f)$ è vera se esiste almeno un elemnto del dominio di x che rende vera f. Analogamente, una formula della forma $\forall x(f)$ è vera se tutti gli elementi del dominio di x rendono f.

Interpretazione logica del calcolo dei domini
Un'espressione del calcolo sui domini $\{A_1 : x_1, ..., A_k : x_k | f\}$ può essere sostanzialmente interpretata come una formula logica del tipo:

$$\{x_1, ..., x_k | f(x_1, ..., x_l)\}$$

dove $x_1, ..., x_k$ sono variabili o costanti, e $f(x_1, ..., x_k)$ è un predicato (vero o falso) che determina l'appartenenza o meno della n-upla di variabili al risultato dell'espressione.

Vantaggi e svantaggi del calcolo dei domini
Il vantaggio principale del calcolo dei domini è la dichiaritività: si *dichiarano*

le proprietà del risultato desiderato, senza specificare i passaggi necessari a calcolarli. I difetti stanno invece nella grande verbosità (non è permesso fare proiezioni su stadi intermedi, ergo siamo costretti a portarci dietro tutti gli attributi nell'enunciato della formula). Inoltre, il calcolo dei domini è dipendente dal dominio (ovvero non tutte le espressioni sintatticamente valide nel calcolo dei domini sono indipendenti dal dominio):

Indipendenza dal dominio

Si dice che un'espressione di un linguaggio d'interrogazione è indipendente dal dominio quando il suo risultato non varia al variare del dominio rispetto alla quale l'espressione su cui viene valutata, salvo ovviamente i valori presenti nell'istanza e nell'espressione. A questo punto un linguaggio si dice indipendente dal dominio se tutte le sue espressioni sono tali. L'algebra relazionale è indipendente dal dominio: i risultati vengono costruiti sulla base di operazioni svolte sulle relazioni presenti nella base di dati senza mai fare riferimento ai domini dei valori che compongono le n-uple delle istanze. In altre parole, tutti i valori compaiono in istanze effettive di relazioni della base di dati o nella formulazione dell'espressione. Il calcolo dei domini, al contrario, è dipendente dal dominio. Si può ad esempio scrivere, senza violare nessuna regola sintattica:

$$\{A : x, B : y | R(A : x) \wedge y = y\}$$

nel risultato, il valore x dell'attributo A è legato alla relazione R, mentre il valore y può essere qualsiasi valore del dominio dell'attributo B, in quanto $\forall B : y | y = y$. Al variare del dominio di B, il risultato dell'espressione cambia (l'espressione è dipendente dal dominio!). Fosse stato B infinito, l'espressione avrebbe restituito un insieme infinito. Un'altro esempio può essere:

$$\{A : x | \neg R(A : x)\}$$

Tale espressione ottiene esattamente il complemento dei valori x sull'attributo A contenuti nella relazione R, ovvero tutti i valori del dominio di A meno quelli in R.

Per ovviare ai problemi di dipendenza dal dominio presentati dal calcolo dei domini è stato introdotto il calcolo delle tuple.

Calcolo delle tuple

Il calcolo delle tuple risolve i problemi di verbosità del calcolo dei domini, utilizzando variabili per denotare tuple anzichè singoli valori legati ad attributi. Ogni relazione coinvolta presenta quindi una variabile, che rappresenta ogni possibile tupla di quella relazione. Per accedere ai singoli attributi di una tupla occorre quindi associare una struttura ad ogni variabile. Le espressioni nel calcolo delle tuple hanno forma:

$$\{T | L | f\}$$

dove:

- T è una **target list** (obiettivi dell'interrogazione);
- L è una **range list**;
- f è una formula.

Target list
Stabiliamo che:

- x è una variabile;
- X è un insieme di attributi di una relazione;
- Y e Z sono sottoinsiemi di attributi di X di pari lunghezza.

A questo punto, possiamo dire che T è una lista di elementi del tipo:

- $Y : x.Z$, Denotiamo con Y solo gli attributi Z della variabile (tupla) x, che assumerà valori definiti in L (*range list*).
- $x.Z \equiv Z : x.Z$ vogliamo solo gli attributi Z della variabile x, che assumerà valori definiti in L, ma non li ridenominiamo.
- $x.* \equiv X : x.X$ Prendiamo tutti gli attributi della variabile x, che assumerà valori definiti in L.

Range list
La *range list* L è una lista che contiene, senza ripetizioni, tutte le variabili della *target list*, con associata la relazione da cui viene prelevata ogni variabile:

$$L \equiv x_1(R_1), ..., x_k(R_k)$$

dove x_i è una variabile e R_i è una relazione. L è una dichiarazione di range, ovvero specifica l'insieme dei valori che possono essere assegnati alle variabili. Non occorrono, come occorrevano nel calcolo dei domini, condizioni atomiche che vincolano una tupla ad appartenere ad una relazione $(A_1 : x_1, ..., A_n : x_n)$.

Formule atomiche del calcolo delle tuple
Possiamo definire, come avevamo fatto per il calcolo dei domini, 2 formule atomiche (manca la condizione di vincolo d'appartenenza alla relazione):

- **Operatori fra variabili**

  $$x_i.A_1 \; OP \; x_j.A_j$$

  dove x e y sono variabili, A_1 e A_j attributi di tali variabili, e OP è un operatore di confronto (uguale, non uguale, maggiore, minore e varianti strette).

- **Operatori fra variabili e costanti**

$$x_i.A_i \; OP \; c, \quad c \; OP \; x_i.A_i$$

  dove c è una costante nel dominio dell'attributo A_i di x_i.

Possiamo quindi definire alcune proprietà delle formule, quasi del tutto analoghe a quelle definite per il calcolo dei domini:

- Le formule atomiche sono formule.
- Se f è una formula, allora anche $\neg f$ è una formula.
- Se f_1 e f_2 sono formule, allora anche $f_1 \wedge f_2$ e $f_1 \vee f_2$ sono formule.
- Se f è una formula e x una variabile che indica una n-upla sulla relazione R, allora anche $\exists x R(f)$ e $\forall x R(f)$ sono formule, dove $\exists$ and $\forall$ sono qualificatori (rispettivamente esistenziale e universale). Notiamo che anche i quantificatori contengono adesso delle dichiarazioni di range(*range list*). La notazione $\exists(R)(f)$ significa "esiste nella relazione R una n-upla x che soddisfa la formula f".

Vantaggi e svantaggi del calcolo delle tuple Il vantaggio principale del calcolo delle tuple è la minore verbosità rispetto al calcolo dei domini. Svantaggiosa è invece l'effettiva impossibilità di esprimere alcune interrogazioni, in particolare l'unione:

$$R_1(AB) \cup R_2(AB)$$

Questo perchè ogni variabile nel risultato ha un solo range, mentre l'unione comporta la derivazione di n-uple da più relazioni distinte. Per questo motivo linguaggi quali l'SQL, definiscono un operatore esplicito di unione (l'UNION). Intersezione e differenza sono invece esprimibili.

Calcolo e algebra

Calcolo e algebra sono sostanzialmente equivalenti, ovvero ogni espressione dell'algebra (indipendente dal domino!) ha un equivalente nel calcolo e viceversa. Esistono però alcune interrogazioni non esprimibili:

- **Calcolo dei valori derivati:** abbiamo la possbilita di estrarre valori, mai di generarne di nuovi.
- **Ricorsività:** le interrogazioni non hanno la possibilità di esprimere ricorsività, come avevamo visto nell'esempio della chiusura transitiva (riguardante riferimenti potenzialmente infiniti fra relazioni).