

Algebra e calcolo relazionale

Ripasso

Le 7 Virtù del DBMS

persistenza
affidabilità
volume
condivisione
riservatezza
efficienza
efficacia

I 4 Livelli di astrazione

Livello fisico (o interno)
Livello logico (aggiunta del significato al livello fisico, senza dettagli implementativi)
Livello esterno (altri modi per vedere un unico livello logico)
Livello concettuale (per modellare il mondo reale)

Le Tabelle

Tabella è una struttura bidimensionale dove ogni colonna è caratterizzata da un nome univoco, ed ogni riga rappresenta una n-upla
Tutte le relazioni matematiche possono essere rappresentate come tabelle
Non tutte le tabelle sono rappresentazioni di relazioni matematiche
Una tabella rappresenta una relazione se

- le righe sono diverse fra loro
- le intestazioni delle colonne sono diverse tra loro
- i valori di ogni colonna sono fra loro omogenei

Valori Nulli

I vincoli:

- intrarelazionali (relativi ai valori di una singola ennupla o del singolo record)
 - di dominio (che riguardano i valori ammissibili per un certo attributo)
 - di ennupla (che legano tra loro i valori dei vari domini di una ennupla)
- relazionali (che coinvolgono una relazione nella sua interezza)
- interrelazionali (cioè tra ennuple di diverse relazioni)

Chiave di una relazione

- concetto di superchiave
- definizione matematica di chiave

La definizione del Modello dei Dati come elemento centrale del DBMS

Algebra e calcolo relazionale

Introduzione ai linguaggi

Il DBMS è una scatola nera, ma dobbiamo interagire con esso.

Per interagire con il DBMS vengono definiti dei linguaggi per le Basi di Dati

Che tipo di operazioni possono essere fatte sul DBMS ?

- Interrogazione sui dati (Query)
- Manipolazione dei dati
- Definizione del modello dei dati
- Regole di sicurezza

Funzioni dei linguaggi:

Interrogazione e manipolazione dei dati :

DML: data manipulation language

select

insert

update

delete

....

TCL: transaction control language

commit

rollback

...

Definizione del modello dei dati :

DDL: data definition language

create

alter

drop

...

Assegnare o revocare permessi di accesso e manipolazione:

DCL: data control language

grant

revoke

....

Tipologie dei linguaggi:

- Dichiarativi - specificano le proprietà del risultato ("che cosa")
- Procedurale - specificano le modalità di generazione del risultato ("come")

Esempi di linguaggi di interrogazione per DB:

- Algebra relazionale – matematico di tipo procedurale basato sull'insiemistica
- Calcolo relazionale – teorico matematico di tipo dichiarativo
- SQL (Structured Query Language) – reale e solo parzialmente dichiarativo
- QBE (Query by Example) – reale di tipo dichiarativo
- Datalog – reale di programmazione logica basato su regole ed inferenza

Insiemistica (o Teoria degli Insiemi)

Vedi presentazione a lato

L'Algebra relazionale

L'Algebra relazionale è una formulazione rigorosa e matematica che permette di operare sulle relazioni, dove le relazioni vengono viste come insiemi di n-uple manipolate attraverso operatori appunto insiemistici.

Si basa su:

- una serie di operatori insiemistici
- che operano sulle relazioni
- che possono essere tra loro combinati
- e che producono come risultato delle altre relazioni

Unione, intersezione, differenza

Sono le tre operazioni base dell'insiemistica ed hanno una corrispondenza diretta con gli operatori della logica booleana.

Sono operazioni binarie, cioè si applicano ad una coppia di relazioni (o insiemi)

Si possono applicare solo a relazioni che hanno lo stesso insieme di attributi o domini

Questa limitazione è di tipo matematico ma anche logico: non si sommano mele con pere

Il risultato di queste operazioni è sempre una relazione il cui schema è a sua volta uguale a quello delle due (o più) relazioni di partenza.

Unione : tutti gli elementi di un insieme o dell'altro – **OR - U**

Intersezione : tutti gli elementi di un insieme e dell'altro – **AND - \cap**

Differenza : tutti gli elementi di un insieme che non sono anche dell'altro - **NOT - -**

La differenza non è la negazione booleana !!!

Queste operazioni possono essere tra loro liberamente combinate ed esistono diversi modi per ottenere lo stesso risultato, seguendo le varie proprietà associative, commutative, etc. ad esempio:

$$\mathbf{r} \cap (\mathbf{s} \cup \mathbf{t}) = (\mathbf{r} \cap \mathbf{s}) \cup (\mathbf{r} \cap \mathbf{t})$$

In particolare l'intersezione può essere definita a partire dalla differenza :

$$\mathbf{r} \cap \mathbf{s} = \mathbf{r} - (\mathbf{r} - \mathbf{s})$$

Quindi l'insieme minimo di operandi è dato da Unione e Differenza.

Cardinalità

La cardinalità è il numero di elementi di un insieme

$|R|$ = numero degli elementi dell'insieme R

Selezione

La selezione è un operatore “monadico”, cioè che si applica ad una sola relazione

Produce una relazione il cui schema è identico a quello della relazione di partenza (detta “operando”)

La relazione ottenuta tramite una **selezione** contiene un sottoinsieme delle n-uple dell'**operando**, in particolare tutte e sole quelle che soddisfano una **condizione** definita dalla selezione stessa.

La selezione ha quindi tre elementi:

1. operando (relazione di partenza)
2. risultato (relazione risultante)
3. condizione (regola o funzione booleana che debbono soddisfare le n-uple dell'operando per appartenere alla relazione risultante)

Sintassi della selezione: $SEL_{Condizione}(Operando)$

Una selezione conterrà al massimo lo stesso numero di n-uple dell'operando, ma tendenzialmente meno in funzione del livello di “ristrettezza” delle condizioni applicate.

Proiezione

La selezione è anche lei un operatore “monadico”.

Produce una relazione il cui schema è un sottoinsieme di quello della relazione di partenza (detta “operando”)

La relazione ottenuta tramite una **proiezione** contiene al massimo un numero di n-uple pari a quello delle n-uple dell'operando.

La proiezione ha tre elementi:

1. operando (relazione di partenza)
2. risultato (relazione risultante)
3. lista degli attributi dell'operando che devono appartenere alla relazione risultante

Sintassi della proiezione: $\text{PROJ}_{\text{ListaAttributi}}(\text{Operando})$

E' importante fare delle considerazioni sulla cardinalità del risultato di una proiezione.

ATTENZIONE:

parliamo di relazioni MATEMATICHE, quindi che NON contengono n-uple uguali.

La relazione ottenuta tramite una **proiezione** contiene al massimo un numero di n-uple pari a quello delle n-uple dell'operando, ma può contenerne meno se:

la lista degli attributi della proiezione non è una superchiave dell'Operando

Se X è superchiave di R, allora numerosità ($\text{PROJ}_X(R)$) = numerosità (R)

Cioè

$$| \text{PROJ}_X(R) | = | R | \text{ se } X \text{ superchiave di } R$$

Considerazioni su Selezione e Proiezione

La selezione e la proiezione possono essere viste come due operazioni tra loro “ortogonali”.

Infatti :

- la selezione “taglia” delle fette orizzontali di una tabella (appunto seleziona delle n-uple tra tutte quelle della relazione di partenza)
- la proiezione “taglia” delle fette verticali della stessa tabella (selezionando solo alcune delle colonne)

La selezione e la proiezione possono essere combinate per ottenere delle porzioni particolari di una tabella (o relazione)

Ridenominazione

La ridenominazione è anche esso un operatore “monadico”, che lascia inalterato il contenuto di una relazione ma ne modifica lo schema cambiando il nome di alcuni attribuiti.

La funzione della ridenominazione è quella di permettere di trasformare due relazioni che hanno informazioni tra loro coerenti ma schemi i cui nomi differiscono, in modo da poter poi applicare su tali relazioni le altre operazioni dell'algebra relazionale.

E' importante quando due o più relazioni esprimono particolari tipologie di un'unica entità che viene in esse dettagliata secondo maggiori dettagli trascurabili nell'interrogazione in atto.

La sintassi di una ridenominazione è: $REN\ y \leftarrow x\ (R)$

Join

Il join (combinazione, aggregazione) è uno degli operatori più interessanti dell'algebra relazionale poiché permette di correlare dati tra relazioni diverse.

$R_1 JOIN R_2$ è una relazione su $X_1 X_2$

$$\{ t \text{ su } X_1 X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2 \\ \text{con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}$$

Cardinalità del JOIN

Il join di R_1 e R_2 contiene un numero di n-uple compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$

INNER JOIN

Join stretto, che implica corrispondenze esatte tra n-uple delle tabelle

NATURAL JOIN

Il NATURAL JOIN è un tipo di operazione che ci permette di correlare due o più tabelle sulla base di valori uguali in attributi contenenti lo stesso tipo di dati ed aventi lo stesso nome.

EQUI JOIN – (Theta Join)

È un tipo di join in cui le righe delle tabelle vengono combinate solo se i campi collegati con join soddisfano una determinata condizione.

OUTER JOIN - Esterno

Con l'OUTER JOIN è possibile estrapolare anche quei dati, *appartenenti ad una delle tabelle*, che non verrebbero estrapolati nei tipi di join visti fino a questo momento. Infatti OUTER significa esterno; dati esterni al normale tipo di join.

Il join esterno completa, con valori nulli, le n-uple che verrebbero tagliate fuori da un join (interno).

Può essere di tre tipi:

- sinistro
- destro
- completo

Join e proiezioni

Prodotto cartesiano

theta-join

$R_1 \text{ JOIN}_{\text{Condizione}} R_2$

Equivalenza di espressioni

Push selections (se A è attributo di R_2)

$\text{SEL } A=10 (R_1 \text{ JOIN } R_2) = R_1 \text{ JOIN SEL } A=10 (R_2)$

Selezione con valori nulli

Viste (relazioni derivate)

Viste virtuali e materializzate

Viste e aggiornamenti

Conclusione Algebra relazionale

Operatori dell'algebra relazionale

unione, intersezione, differenza

selezione

proiezione

ridenominazione

join (join naturale, prodotto cartesiano, theta-join)

Calcolo relazionale

Il Calcolo relazionale fa parte di una famiglia di linguaggi dichiarativi, basati sul calcolo dei predicati del primo ordine.

Calcolo su Domini

Questo tipo di calcolo è composto da espressioni applicate a domini.

Le espressioni hanno la forma:

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

dove:

f è una formula (che può utilizzare connettivi booleani e quantificatori)

$A_1: x_1, \dots, A_k: x_k$ è una lista di coppie attributo – variabile, detta “target list”, con

A_1, \dots, A_k attributi distinti (che possono anche non far parte della base di dati)

x_1, \dots, x_k variabili tra loro distinte

Il significato o risultato di un'espressione è una relazione sui domini A_1, \dots, A_k che contiene le n-uple di valori per x_1, \dots, x_k tali per cui la formula f risulta verificata

Nelle formule possono essere utilizzati i qualificatori universali ed esistenziali:

- per ogni \forall
- esiste \exists

Il calcolo sui domini ha il difetto di essere molto lungo (per scrivere una semplice interrogazione ci vogliono predicati molto lunghi, e quindi è di difficile lettura) e soprattutto permette di scrivere espressioni senza senso.

Calcolo su ennuple con dichiarazioni di range

Per superare i limiti del calcolo sui domini viene definito il “calcolo su ennuple con dichiarazioni di range”, dove per semplificare e snellire la scrittura ci si può riferire ad un'intera n-upla utilizzando una singola variabile (invece della lista delle coppie attributo-variabile), ed i valori ed i domini vengono ristretti ai soli provenienti da una definita base di dati.

Le espressioni del “calcolo su ennuple con dichiarazioni di range” hanno la seguente forma:

$$\{ TargetList \mid RangeList \mid Formula \}$$

dove

- *RangeList* è la lista delle relazioni associate alle relative variabili d'uso
- *TargetList* è la lista degli attributi che verranno estratti dalla *RangeList*
- *Formula* è appunto una formula che deve essere soddisfatta dalle n-uple delle relazioni della *RangeList* affinché i loro valori entrino a far parte del risultato nella *TargetList*

Calcolo e algebra

Calcolo e algebra sono "equivalenti"

per ogni espressione del calcolo relazionale che sia indipendente dal dominio esiste un'espressione dell'algebra relazionale equivalente a essa

per ogni espressione dell'algebra relazionale esiste un'espressione del calcolo relazionale equivalente a essa (e di conseguenza indipendente dal dominio)

Limiti dell'algebra e del calcolo relazionale

Chiusura transitiva

Supervisione(Impiegato, Capo)

Per ogni impiegato, trovare tutti i superiori (cioè il capo, il capo del capo, e così via)

Nell'esempio, basterebbe il join della relazione con se stessa, facendo però ad ogni passo un'opportuna ridenominazione, ma servirebbero potenzialmente infinite join e ridenominazioni

Chiusura transitiva, impossibile!

Non esiste in algebra e calcolo relazionale la possibilità di esprimere l'interrogazione che, per ogni relazione binaria, ne calcoli la chiusura transitiva