ALGEBRA RELAZIONALE E GESTIONE DEI VALORI NULL

APPUNTI DI ROBERT PARCUS BETOPARCUS@GMAIL.COM

2012/10/16 - Algebra relazionale

1. La divisione

Indicato piú intuitivamente controparte della quantificazione universale.
Esempio: Date due relazioni

ISCRIZIONE:

matricola	id_corso
123	BD
283	BD
123	INF
123	MAT
283	MAT
375	BD
283	INF

CORSO:

id_corso
BD
INF
MAT

$R \leftarrow ISCRIZIONE \div CORSO$
(mi dará le matricole iscritte a tutti i corsi appartenenti a CORSO)

R:

123	283

Vediamo come si puó derivare la divisione dalle altre operazioni che conosciamo:

Definizione $[DIVISIONE]$: Siano $R1(x), R2(y)$ due schemi di relazione t.c. Y appartiene a X e siano $r1, r2$ due istanze di $R1R2$. L'operatore divisione produce una relazione le cui tuple, se estese con una qualunque tupla della seconda relazione producono una tupla di $R1$.
$r1 \div r2 = \{\, t \,|\, \forall\, t' \in r2,\ t \cup t \in r1 \,\}$

La divisione $r1 \div r2$ é dunque un operatore definito sullo schema con attributi $X - Y$. E' un operatore derivato e puó essere definito come:
$T1 \leftarrow \Pi_{x-y}(R1)$
$T2 \leftarrow \Pi_{x-y}(R2 \times T1) - R1)$

Sull'esempio di prima:
$T1 \leftarrow \Pi_{matricola}(\mathit{Iscrizione})$
$T2 \leftarrow \Pi_{matricola}((T1 \times \mathit{corso}) - R1)$
$R \leftarrow \mathit{T1T2}$

Date: 10 ottobre 2012.

2. ALGEBRA RELAZIONE E VALORI NULL

La presenza di valori nulli nelle istanze di una base di dati richiede un'estensione della semantica degli operatori dell'algebra relazionale. L'approccio tradizionale (usato anche nei DBMS commerciali ed in SQL) é quello di considerare una logica a 3 valori per la valutazione delle formule proposizionali e quei nodi degli operatori di Selezione e Join.

Proiezione, unione, differenza con valori nulli:

Continuano a comportarsi usualmente. L'uguaglianza tra NULL é un livello sintattico e due tuple sono sono uguali anche se ci sono valori NULL.

Esempio:

IMPIEGATI:

codice	nome	ufficio
123	Rossi	A12
231	Verdi	NULL
373	Verdi	A27
435	Verdi	NULL

Responsabili:

codice	nome	ufficio
123	Rossi	A12
NULL	NULL	NULL
435	Verdi	A27

manca roba...

Selezione e valori nulli:

Per la selezione, il problema é stabilire se in presenza di valori NULL un predicato é vero o meno.

IMPIEGATI:

codice	nome	ufficio	risultato
123	Rossi	A12	OK
231	Verdi	NULL	???
373	Verdi	A27	OK

$\Pi_{ufficio='A12'}(IMPIEGATI)$

La seconda tupla fa parte del risultato?
Non si puó sapere.
Vanno definite politiche di gestione del NULL:

Per verificare predicati come quelli dell'esempio precedente, si introduce una logica a 3 valori, dove oltre al valore V ed F si ha un terzo valore '?'.

a=a	V
a=b	F
a=NULL	?
NULL=NULL	?
a!=a	F
a!=b	V
a!=null	?
null!=null	?

Queste tabelle di veritá possono essere estese ad altri tipi di confronto,. Se D é un dominio su cui é definito un ordinamento $<$ $(>, >=, <=)$, allora per 'e' appartenente $<, >=$ il confronto x e y é indefinito se x o y sono NULL. Le tabelle di veritá nella logica a 3 valori per gli operatori booleani sono:

NOT:

V	F
F	V
?	?

AND:

V	V F ?
F	F F F
?	? F ?

OR:

V	V V V
F	V F ?
?	V ? ?

Una selezione $G(f)$ produce le sole tuple per cui la condizione di selezione risulti true.
Esempio: $\sigma_{\text{ufficio='A12' OR ufficio != 'A12'}}(IMPIEGATI)$ Non restituisce tutte le tuple!!! Per lavorare esplicitamente con i valori nulli si introducono le condizioni atomiche:

- a IS NULL;
- a IS NOT NULL;

$\sigma_{(ufficio='A12'||ufficio!='A12'||ufficioISNULL)}(IMPIEGATI)$

JOIN CON VALORI NULL:

Valgono le stesse regole di valutazione per F utilizzate nella selezione. Il JOIN NATURALE non combina due tuple se queste hanno entrambe valori nulli su un attributo in comune.

Esempio:

Impiegati:

codice	nome	uffici
123	Rossi	A12
231	Verdi	NULL
272	"	A27
435	"	NULL

Responsabili:

ufficio	codice
A12	123
A27	NULL
NULL	231

3. ESERCIZI 18/10

Consideriamo la seguente porzione di BD:

FORNITORE(<u>CodForn</u>, nome, cittá)
PRODOTTI(<u>CodProd</u>, nome, marca, modello)
CATALOGO(<u>CodForn</u>,<u>CodProd</u>, Costo)

2) Trovare i nomi dei fornitori che distribuiscono prodotti di marca IBM:

$$T_1 \leftarrow \sigma_{marca=IBM}\,(PRODOTTI \bowtie CATALOGO)$$

3)Trovare i codici di tutti i prodotti che sono forniti da almeno 2 fornitori:
<u>Idea:</u> Concateniamo due copie di catalogo su valori uguali dell'attributo codice prodotto e troviamo le tuple su cui i fornitori sono diversi.

$$COPIACATALOGO(codforn1, codforn2) \leftarrow \Pi_{codforn,prod}(CATALOGO)$$
$$T_1 \leftarrow COPIACATALOGO \bowtie CATALOGO$$
$$RIS \leftarrow \Pi(\sigma_{codforn1,codforn2}(T1)$$

4)Nomi dei fornitori che distribuiscono tutti i prodotti:

$$T1 \leftarrow (\Pi_{codFor,codProd}(CATALOGO)) \div (\Pi_{codProd}(PRODOTTI))$$
$$RIS \leftarrow \Pi(T_1 \bowtie FORNITORE)$$

Oppure, senza usare la divisione:
//Tutte le coppie fornitore e prodotto

$$T_1 \leftarrow \Pi_{codforn}(CATALOGO) \times \P_{codProd}(PRODOTTO)$$
$$//CodForn \notin T2secod_F ornforniscetuttiiprodotti.$$
$$T_2 \leftarrow \Pi(T_1 - \Pi_{codForn,codProd}(CATALOGO))$$
$$RIS \leftarrow \Pi_{codForn}(CATALOGO) - T2$$

STADIO(<u>nome</u>, cittá, capienza) PARTITA(<u>stadio, dato, ora</u>, squadra1, squadra2) SQUADRA(<u>nazione</u>, continente, livello)

1) Determinare gli stadi in cui non gioca alcuna squadra africana:

$$AFRICA \leftarrow \Pi_{(nazione)}(\sigma_{(continente \neq africa)}(SQUADRA)))$$
$$BAD \leftarrow \Pi_{(stadio)}(PARTITA \bowtie_{(nazione=squadra1ANDnazione=squadra2)} AFRICA)$$
$$RIS \leftarrow \Pi_{(nome)}(STADIO) - BAD$$

2)Determinare le squadre che incontrano soltanto squadre dello stesso livello:

$$\text{SQUADRA1\_LIVELLO(S1, S2, L1)}$$
$$\leftarrow \Pi_{(squadra1,squadra2,livello)}(PARTITA \bowtie_{(squadra1=nazione)} SQUADRA$$
$$\text{SQUADRA2\_LIVELLO(S1, S2, L1, l2)}$$
$$\leftarrow \Pi_{(S1,S2,L1,livello)}(SQUADRA1\_LIVELLO \bowtie_{(S2=nazione)} SQUADRA$$
$$BAD1 \leftarrow \Pi_{(S1)}(\sigma_{L1 \neq L2}(SQUADRE\_LIVELLO)))$$
$$BAD2 \leftarrow \Pi_{(S2)}(\sigma_{L1 \neq L2}(SQUADRE\_LIVELLO))$$
$$RIS \leftarrow \Pi_{(nazione)}(squadra) - (BAD1 \cup BAD2)$$

Sia dato il seguente schema relazione che descrive gli esami (obbligatori e non) in un anno (I, II, III) di un indirizzo di laurea triennale:

CORSO(<u>cod_c</u>, nome, CFU, SSD)
INDIRIZZO(<u>COD_1</u>, titolo)
COMPOSIZIONE(<u>COD_I, COD_C</u>, tipo, anno)

1)Determinare i titoli dei corsi che in almeno un indirizzo possono essere collocati indifferentemente in ciascuno dei tra anni:

$$ANNO1(codc1, codI1) \leftarrow \Pi_{(codc,codI)}\sigma_{(anno=1)}(COMPOSIZIONE)$$
$$ANNO2(codc2, codI2) \leftarrow \Pi_{(codc,codI)}\sigma_{(anno=2)}(COMPOSIZIONE)$$
$$ANNO3(codc3, codI3) \leftarrow \Pi_{(codc,codI)}\sigma_{(anno=3)}(COMPOSIZIONE)$$
$$GOOD \leftarrow (ANNO1 \bowtie_{(codc1=codc2ANDcodI1=codI2)}$$
$$ANNO2) \bowtie_{(codc2=codc3ANDcodI2=codI3)} ANNO3)$$
$$RISULTATO \leftarrow \Pi_{(TITOLO)}(GODD \bowtie_{(codc1=codc)} CORSO)$$

2)Determinare i titoli degli indirizzi che prevedono lo stesso insieme di esami obbligatori dell'indirizzo 'Sistemi'://

Corsi obligatori dell'indirizzo sistemi

$$OBBSIST \leftarrow \Pi_{(codc)}(\sigma_{(TITOLO=sistemiANDtipo=obb)}(INDIRIZZO \bowtie COMPOSIZIONE))$$

Tutti i corsi obbligatori
$OBB \leftarrow \Pi_{(codI,codc)}(\sigma_{(tipo=obb)}(COMPOSIZIONE))$
indirizzi che non voglio nel risultato
$\Pi incodi((piincodi(indirizzi)xOBBSIST)-OBB) \cup \Pi_{codI}(OBB-(\Pi_{codI}(INDIRIZZI) \times OBBSIST))$

4. Calcolo relazionale sulle tuple (cap. 6.6 libro)

A differenza del linguaggio relazione, il calcolo relazionale e' dichiarativo. *Si dice quello che si vuole.* Il calcolo e' una base anche per il linguaggio SQL, perche' simile. Ha lo stesso potere espressivo dell'algebra relazionale di base. Si parla infatti di un linguaggio completo.
Il calcolo relazionale sulle tuple e' un linguaggio di interrogazione formale per il modello relazionale.

- dichiarativo;
- base per SQL;
- stesso potere espressivo dell'algebra relazionale di base;

DEFINIZIONE:[LINGUAGGIO COMPLETO] Un linguaggio L di interrogazione formale per il modello relazione e' detto completo se possiamo esprimere in L qualsiasi interrogazione esprimibile nel calcolo relazione.

Esempio di calcolo relazionale sulle TUPLE:
Si trovi la data di nascita e l'indirizzo dell'impiegato John Smith.

{ t.DATA_N, t.INDIRIZZO | IMPIEGATO(t) AND t.NOME = John AND t.COGNOME = Smith } {Taget list | Espressione condizionale sulle variabili di tuple}

DEFINIZIONE [ESPRESSIONE DEL CALCOLO RELAZIONALE]- SINTASSI: Un espressione del calcolo relazionale e' un'espressione del tipo { T | a } Dove T e' la target list (liste degli obbiettivi dell'interrogazione). In particolare:

- T e' una lista con elementi del tipo t.A dove t e' una variabile di tuple, A p un attributo della relazione sulle cui tuple prende valore t.
- se t1,...,tn sono le variabili di tipla in T, allora a = cond(t1,...,tn) e' una condizione.

Al fine di definire il concetto di formula del calcolo relazione, introduciamo le nozioni di atomi del calcolo relazionale:

DEFINIZIONINE[ATOMO CALCOLO RELAZIONE] SINTASSI: Un atomo del calcolo relazionale e' una'espressione appartenente ad uno dei 3 tipi di espressioni elencate di seguito:

- R(t) dove R e' un nome di relazione e t e' una variabile di tupla (ex. IMPIEGATO(t));

- $t_i.A\theta t_j.Bdove\theta \in \{=,<,>,<=,>=\}$.
  t_i e' un attributo variabile di tupla ed A e' un attributo della relazione sulle cui tuple prende valore t_i. (risp t_j,B);
- $t_i.A\ \theta$ c dove $\theta\ \in \{=,<,>,<=,>=\}$, t e' una variabile di tupla, c una costante, A ....;

$\{\ T\ |\ a\ \}$ Possiamo definire formalmente le formule del calcolo relazionale:
DEFINIZIONE[FORMULE CALC.REL]SINTASSI:
Una formula (o condizione) del calcolo relazionale e' definita induttivamente come segue 1)ogni atomo del calcolo relazione e' una formula; 2)Se F_1, F_2 sono formule, allora $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \neg F_2$ sono formule; 3) Se F e' una formula allora $\forall\, tF$, $\exists tF$ sono formule;

Seconda parte:

Per poter definire il valore di verita' (semantica) delle formule del calcolo relazionale dobbiamo prima introdurre il concetto di variabile *libera* o *legata* in una formula. Intuitivamente, una variabile di tupla e' legata se quantificata.
Formalmente, un'occorrenza di variabile di tupla t viene definita libera/legata nelle formule del calcolo relazionale F in base alle seguenti regole:

- Se F e' un atomo, ogni occorrenza di t libera in F;
- Se $F \in \{F_1 \wedge F_2, F_1 \wedge F_2, \neg F_1\}$ t libera in F se t e' libera nella sottoforma $F_1 F_2 incuicompare$
- tutte le occorrenze di variabili t libere in F sono legate in $\exists t.F, \forall t.F$

ESEMPIO:
$F_1 : d.NOME\_D =' Ricerca' \rightarrow de'liberainF_1\ F_2 \exists t(d.numero = N\_D) \rightarrow de'liberoinF_2$
$F_3 : \forall d(\exists t(d.numero = t.N\_D)) \rightarrow de'legatainT_3$
DEFINIZIONE [FORMULE CALC REL] SEMANTICA
Il valore di verita' di una formula del calcolo relazione data dalle seguenti regole:
1) R(t) vera se t assegnata ad una tupla di R;
2) $t_i \wedge \theta t_2 Bveraseti, tjprendonovalorisutupletalicheivaloridegliattributispecificatistannoinrealzioni\theta$;
3) $F \in \{F_1 \wedge F_2, F_1 \wedge F_2, \neg F_1\}$ hanno l'usuale significato;
4) $\exists t.F$ ha valori di verita' vero se il valore della formula F e' vero per almeno una tupla assegnata ad occorrenze di t libere in F
5) $\forall T.F$ e' vero se F e' vera $\forall$ tupla assegnata ad occorrenze libere di t in F

Un'espressione $\{T|C\}$t C del calcolo relazione restituisce la tupla (rispetto agli attributi nella target list) in cui a assume valore vero.

ESERCIZI:
1) Trovare nome congnome degli impiegati che lavorano al dipartimento di ricerca:

{ t.NOME, t.COGNOME|IMPIEGATO(t) $\wedge$ $\exists$d(DIPARTIMENTO(d)) $\wedge$
d.nome = 'Ricerca ' $\wedge$ $(NUMERO\_D = t.N\_D)$

2) Per ogni progetto con sedi a Stafford si elenchi il numero di progetto, il numero del dipartimento che lo gestisce, l'indirizzo, il nome e le date di nascita del direttore del dipartimento:

{p.NUMERO_P, p.NUM_D, t.COGNOME, t.DATA, t.INDIRIZZO|PROGETTO(t)$\wedge$
IMPIEGATO(t) AND p.SedeP = 'Stafford' AND$\exists$(d(DIPARTIMENTO(d) AND d.SSN_DIR = t.SSN AND

Si noti che:

- variabili libere sono p, t;
- d legata; la condizione C viene valutata per ogni coppia di tuple assegnata a p e t ovvero per ogni coppia di tuple assegnate a p e t si verifica che: 1) t sia un tupla di impiegato e p di progetto 2) se il progetto ha sedi a Stafford; 3) Se esiste un dipartimento il cui numero c MANCA ROBA (ha cancellato!!!)