

Algebra relazionale

Wednesday, 26 April 2023

10:41

Differenze AR, SQL:

AR	SQL
Linguaggio formale, permette di comprendere le operazioni SQL Aka è la base dei linguaggi reali, se la impariamo bene SQL sarà facile	Linguaggio in sè per sè
Istruzioni equivalenti possono differire in efficienza	Istruzioni equivalenti hanno la stessa efficienza diversa leggibilità
Le relazioni sono intese in senso matematico	Relazioni sono intese come tabelle
No righe uguali siccome in un insieme non esistono elementi duplicati	Possono esistere righe duplicate

- Insieme di operatori:

- Relazioni
- Relazioni che producono relazioni

E possono essere:

- Unarie

Operano su 1 sola relazione

- Selezione
 - Prendiamo tutti quelli che soddisfano una condizione
 $\sigma_{Ruolo='Associato'}(Personale_{docente})$
- Proiezione
 - Considerare un sottoinsieme degli attributi di una relazione
 $Proj_{lista-attributi}(r)$
In lista-attributi mettiamo gli attributi che desideriamo
- Ridenominazione
 - Cambia il nome di un attributo
 $REN_y < -_x(r)$
Cambia il nome dell'attributo x della relazione r in y
Nota: questa modifica è momentanea, solo all'interno della query, poi è cancellata
nulla fosse successo
 - $\sigma_{Ruolo='Associato'}(Personale_{docente})$ \rightarrow $\sigma_{Ruolo='Associato'}(Studente)$

- $r_docente \cup r_non_docente \cup rel_matricola_d \searrow \neg matricola_st \setminus studente$
 Noi qui stiamo dando studente con matricola_st -> matricola_d
 Però, è come se creassimo una copia che viene cancellata finita la query
 Quindi studente non viene toccato

- Binarie

2 relazioni che produce 1 relazione

- Unione (mette uno sopra l'altro)
 - Date due relazioni r1, r2
 $r_1 \cup r_2$
 Produce tutte le n-ple delle relazioni r1, r2 (ordinate)
 - Dato personale_docente(cognome,nome), personale_non_docente(cognome,nome)
 Formulare una relazione di algebra relazionale che comprende tutti i dipendenti dell'ateneo
 L'impse ci richiede il numero totale di personale.
 Per averlo dobbiamo fare
 $Personal_Docente \cup Personal_Non_Docente$
 - Nota: siccome non possiamo avere righe duplicate, la cardinalità potrebbe essere meno della somma delle cardinalità
- Intersezione
 - Date due relazioni r1, r2
 $r_1 \cap r_2$
 Produce una relazione che contiene ennuple che contengono ad entrambi
 - Abbiamo personale_docente, non_docente, studente
 Noi vogliamo sapere i non_docenti che sono anche studenti
 $non_docente \cap studente$
- Differenza
 - $r_1 - r_2$
 R1 senza r2
- Join (mette uno a sinistra dell'altro)
 - Combiniamo tuple appartenenti a relazioni diverse
 - 2 tipologie che ci interessano:
 - Join naturale
 $R1(X1) \bowtie R2(X2)$
 Controllo di tutti con lo stesso nome
 - Theta join
 Un join con delle condizioni
 $Pers \bowtie_{classe_stipendio=Classe_Stipendio}$
 - Tuple dangling sono tuple che vengono scartate dal join
 - Ed i join possono essere:
 - Interno

Potremmo non avere valori null, ci possono essere delle tuple scartate

□ Esterno

Potremmo avere valori nulli, e nel caso delle tuple dovrebbero essere scartate, vengono comunque inserite con dei campi null

E qui abbiamo 3 tipologie:

- ◆ Left join, quindi le tuple scartate a sinistra vengono aggiunte con valori nulli
- ◆ Right join, come prima solo a destra
- ◆ Full, tutti e due di sopra

Si aggiunge left/right/full come condizione nel join

- Ed in più un join si può chiamare "self" quando facciamo un join con se stesso
Utile quando vogliamo confrontare due righe della stessa tabella

E ricorda che le relazioni sono insiemi, siccome sono il prodotto cartesiano degli insiemi

$A \times B$ = insieme di n-uple, aka insieme di coppie

[Inserire foto prodotto cartesiano è un insieme di n-uple]

Le tuple sono ordinate

- Viste

Sono delle query che ci teniamo in memoria: Nome vista = espressione AR

Praticamente semplifichiamo le query spezzettandole

Nota:

- Strategia Pushing selection down

Per questioni di efficienza, è sempre meglio fare prima tutte le selezioni, e poi le altre operazioni

- Algebra relazionale è case sensitive

- Se abbiamo un NULL, allora qualunque operazione è ???

Se vogliamo prendere cose nulla, dobbiamo dire "IS NULL"

Dire " $X = 2100$ or $X \neq 2100$ " ci darà tutti i valori non nulli

- Facendo il join, se ci sono 2 NULL essi non verranno presi: nel join solo se diversi da NULL

Quando usale quale query?

Seguire questi passi:

- 1) Identificare dove sono le informazioni che ci servono, aka quali attributi e quali tabelle
E se sono coinvolte più tabelle, comprendere se un join è necessario
- 2) Comprendere se dobbiamo fare operazioni insiemistiche, aka unioni
- 3) Ora che abbiamo 1 unica tabella con tutte le soluzioni, facciamo eventuali selezioni per filtrare le righe
- 4) Facciamo una proiezione per tenere solo le colonne che ci interessano