Algebra relazionale

- Il linguaggio per interrogare ed aggiornare la base di dati fa parte del modello di questi ultimi.
- Un’operazione di aggiornamento può essere vista come una funzione che, data un’istanza di un database, ne produce un’altra. Analogamente un’interrogazione può essere vista come una funzione che, data un’istanze di un database, produce una relazione.
- L’algebra relazionale è un calcolo procedurale, i.e., le funzioni sono specificate per mezzo della descrizione delle operazioni da svolgere per ottenere il risultato.
- L’algebra relazionale consiste di un insieme di operatori che, data una relazione, producono un’altra relazione come risultato. Ciò permette di comporre i vari operatori in modo da formare espressioni complesse (query).

Tipi di operatori

- Gli operatori dell’algebra relazionale sono classificabili nelle seguenti categorie:
  - Operatori insiemistici (unione, intersezione, differenza)
  - Operatori di rinomina, selezione e proiezione
  - Operatori di join (naturale, prodotto cartesiano e theta-join)
- Gli operatori insiemistici sono naturalmente applicabili in quanto le relazioni sono degli insiemi di tuple. Tuttavia, affinché unione, intersezione e differenza abbiano senso, queste ultime vanno applicate a relazioni definite sugli stessi attributi. Le tuple infatti devono sempre essere *omogenee* all’interno di una relazione.
- L’operatore di rinomina serve a superare in certi casi lo stretto vincolo precedente, rinominando opportunamente alcuni attributi delle relazioni in gioco, in modo che gli operatori insiemistici possano produrre una relazione corretta.

Operatori insiemistici

- L'unione di due relazioni $r_1(X)$ e $r_2(X)$, definite sugli stessi attributi, si denota con $r_1 \cup r_2$ ed è ancora una relazione definita su X contenente le tuple che appartengono a r_1, a r_2 oppure ad entrambe.
- L'intersezione di due relazioni $r_1(X)$ e $r_2(X)$, definite sugli stessi attributi, si denota con $r_1 \cap r_2$ ed è ancora una relazione definita su X contenente le tuple che appartengono contemporaneamente a r_1 ed a r_2.
- La differenza di due relazioni $r_1(X)$ e $r_2(X)$, definite sugli stessi attributi, si denota con $r_1 - r_2$ ed è ancora una relazione definita su X contenente le tuple che appartengono a r_1, ma non a r_2.
- Si noti che unione ed intersezione sono degli operatori simmetrici ($r_1 \cup r_2 = r_2 \cup r_1$, $r_1 \cap r_2 = r_2 \cap r_1$), mentre la differenza no ($r_1 - r_2 \neq r_2 - r_1$).

Operatore di rinomina

- Sia r una relazione su X e sia Y un altro insieme di attributi con la stessa cardinalità. Siano inoltre $A_1A_2...A_k$ e $B_1B_2...B_k$ due ordinamenti degli attributi in X ed in Y, rispettivamente. La rinomina $\rho_{B_1B_2...B_k \leftarrow A_1A_2...A_k}(r)$ è una relazione contenente una tupla t' per ogni tupla t di r in modo che $t'[B_i]=t[A_i]$ per i=1,…,n.
- La definizione conferma che a cambiare sono soltanto i nomi degli attributi, mentre i valori rimangono inalterati.
- Per convenzione di solito si indicano in X e Y soltanto gli attributi che vengono effettivamente rinominati, i.e., quelli per cui $B_i \neq A_i$.

Operatore di selezione

- Data una relazione r(X), una formula proposizionale F su X è una formula ottenuta combinando con gli operatori logici $\vee$, $\wedge$, $\neg$ degli atomi del tipo A ϑ B o A ϑ c che rispettino i seguenti vincoli:
  - ϑ è un operatore di confronto ($=, \neq, >, <, \geq, \leq$);
  - A e B sono degli attributi in X compatibili, i.e., ha senso eseguire un confronto fra i valori dei rispettivi domini;
  - c è una costante compatibile con il dominio di A.
- Data una formula proposizionale F su X ed una tupla t su r(X), il valore di verità di F su t è definito come segue:
  - A ϑ B è vera su t se e solo se t[A] è in relazione ϑ con t[B];
  - A ϑ c è vera su t se e solo se t[A] è in relazione ϑ con c;
  - $F_1 \vee F_2$ è vera su t se e solo se è vera F_1 oppure è vera F_2 oppure sono vere entrambe;
  - $F_1 \wedge F_2$ è vera su t se e solo se è vera sia F_1 che F_2;
  - $\neg$F è vera su t se e solo se non è vera F;
- La selezione $\sigma_F(r)$ produce come risultato una relazione composta dalle tuple di r per cui F è vera.

Operatore di proiezione

- Data una relazione r(X) ed un sottoinsieme Y di X ($Y \subseteq X$), la proiezione di r su Y si denota con $\pi_Y(r)$ e consiste nell'insieme di tuple su Y ottenute dalle tuple di r prendendo in considerazione soltanto i valori su Y: $\pi_Y(r)=\{t[Y] \mid t \in r\}$.
- In pratica tutte le tuple contribuiscono al risultato di una proiezione, ma soltanto con i valori corrispondenti agli attributi specificati dall'operatore.
- Siccome una proiezione riduce il numero di attributi che andranno a caratterizzare la relazione risultato, è possibile che quest'ultima contenga meno tuple dell'operando originale. Infatti le tuple che assumono valori uguali per gli attributi inclusi in Y daranno un unico contributo complessivamente nella relazione risultato.
- In generale $\pi_Y(r)$ avrà la stessa cardinalità di r se e soltanto se Y è una superchiave per r.

L’operatore join

- L’operatore join è il più importante dell’algebra relazionale in quanto permette di stabilire delle connessioni fra dati contenuti in relazioni differenti. Ciò avviene confrontando i valori in esse contenute, seguendo le caratteristiche del modello relazionale che è *value based*.
- Esistono due versioni principali di join (definibili l’una a partire dall’altra e viceversa).
- La prima versione (natural join) è utile da un punto di vista teorico ed introduttivo.
- La seconda versione (theta join) è maggiormente utile dal punto di vista pratico ed è quella comunemente implementata in SQL.

Natural join

- Date due relazioni $r_1(X_1)$, $r_2(X_2)$, il loro natural join $r_1 \bowtie r_2$ è una relazione su X_1X_2 definita come segue:
  $r_1 \bowtie r_2 = \{t \text{ su } X_1X_2 \mid \text{esiste } t_1 \in r_1,\ t_2 \in r_2 \text{ tali che } t[X_1]=t_1 \text{ e } t[X_2]=t_2\}$
  o, equivalentemente, $r_1 \bowtie r_2 = \{t \text{ su } X_1X_2 \mid t[X_1] \in r_1 \text{ e } t[X_2] \in r_2\}$.
- Quindi il risultato è ottenuto combinando le tuple di r_1 e r_2 che assumono gli stessi valori in corrispondenza degli attributi in comune.
- Il grado del risultato sarà minore od uguale alla somma dei gradi dei due operandi.
- La cardinalità del risultato è soggetta alle regole seguenti:
  - Se il join di r_1, r_2 è completo (i.e., ogni tupla di r_1 e di r_2 contribuisce ad almeno una tupla del risultato), allora il numero delle tuple finali sarà maggiore od uguale al massimo fra $|r_1|$ e $|r_2|$.
  - Se $X_1 \cap X_2$ contiene una chiave di r_2, allora il numero delle tuple finali sarà minore od uguale a $|r_1|$.
  - Se $X_1 \cap X_2$ è la chiave primaria di r_2 ed esiste un vincolo referenziale fra $X_1 \cap X_2$ in r_1 e tale chiave di r_2, allora il numero delle tuple finali sarà esattamente uguale a $|r_1|$.

Outer join

- L'outer join (nelle sue varianti left, right e full) permette di evitare che vengano lasciate fuori le tuple che non hanno un corrispondente nell'altra relazione.

r_1

Impiegato	**Dipartimento**
Smith	Vendite
Black	Produzione
White	Produzione

r_2

Dipartimento	**Capo**
Produzione	Mori
Acquisti	Brown

Impiegato	Dipartimento	Capo
Smith	Vendite	NULL
Black	Produzione	Mori
White	Produzione	Mori

$r_1 \bowtie_{LEFT} r_2$

Impiegato	Dipartimento	Capo
Black	Produzione	Mori
White	Produzione	Mori
NULL	Acquisti	Brown

$r_1 \bowtie_{RIGHT} r_2$

Impiegato	Dipartimento	Capo
Smith	Vendite	NULL
Black	Produzione	Mori
White	Produzione	Mori
NULL	Acquisti	Brown

$r_1 \bowtie_{FULL} r_2$

Proprietà del join

- Commutatività: $r_1 \bowtie r_2 = r_2 \bowtie r_1$.
- Associatività: $r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3$.
- Risulta quindi possibile definire un join n-ario:

  $r_1 \bowtie r_2 \bowtie \ldots \bowtie r_n$ abbreviabile in $\bowtie_{i=1}^{n} r_i$.
- Quando $X_1 = X_2$ il natural join coincide con l'intersezione fra tuple: $r_1(X_1) \bowtie r_2(X_1) = r_1(X_1) \cap r_2(X_1)$.
- Quando $X_1 \cap X_2 = \emptyset$ il natural join coincide con il prodotto cartesiano dove tutte le tuple di entrambi gli operandi partecipano al join non avendo attributi in comune su cui richiedere l'uguaglianza dei rispettivi valori.

Theta-join ed equi-join

- Il prodotto cartesiano di per sé non è molto utile in quanto combina le tuple di due relazioni in modo non necessariamente significativo; tuttavia diventa interessante quando è seguito da una selezione: $\sigma_F(r_1 \bowtie r_2)$.

Impiegato	Reparto
Rossi	A
Verdi	A
Gialli	B

Impiegati

Codice	Nome
A	Contabilità
B	Produzione

Reparti

Impiegato	Reparto	Codice	Nome
Rossi	A	A	Contabilità
Rossi	A	B	Produzione
Verdi	A	A	Contabilità
Verdi	A	B	Produzione
Gialli	B	A	Contabilità
Gialli	B	B	Produzione

Impiegati $\bowtie$ Reparti

Impiegato	Reparto	Codice	Nome
Rossi	A	A	Contabilità
Verdi	A	A	Contabilità
Gialli	B	B	Produzione

$\sigma_{Reparto=Codice}$(Impiegati $\bowtie$ Reparti)

- La combinazione del prodotto cartesiano e dell'operatore di selezione dà origine all'operatore theta-join che si indica con l'espressione $r_1 \bowtie_F r_2$ (quando F è una congiunzione di atomi del tipo A=B in cui A è un attributo della prima relazione e B un attributo della seconda, si parla di equi-join).

Derivabilità del natural join

- Sì è visto come il theta-join sia un operatore definibile a partire dal prodotto cartesiano (i.e., natural join su due relazioni $r_1(X_1)$ $r_2(X_2)$ dove $X_1 \cap X_2$).
- Si può anche fare il viceversa, i.e., definire il natural join a partire dall'equi-join, utilizzando l'operatore di rinomina e quello di proiezione. Ad esempio, date le due relazioni $r_1(ABC)$ e $r_2(BCD)$, il loro natural join si ottiene come segue:
  - Si rinominano gli attributi di r_2 in modo da ottenere due schemi disgiunti: $\rho_{B'C' \leftarrow BC}(r_2)$.
  - Si esegue l'equi-join $r_1 \bowtie_{B=B' \wedge C=C'} (\rho_{B'C' \leftarrow BC}(r_2))$, ponendo le condizioni di uguaglianza sugli attributi rinominati.
  - Tramite una proiezione si eliminano gli attributi rinominati (che contengono dei dati duplicati):

$$\pi_{ABCD}(r_1 \bowtie_{B=B' \wedge C=C'} (\rho_{B'C' \leftarrow BC}(r_2)))$$

Esempio di query

Impiegati

Numero	Nome	Età	Stipendio
101	Mary Smith	34	40000
103	Mary Bianchi	23	35000
104	Luigi Neri	38	61000
105	Nico Bini	44	38000
210	Marco Celli	49	60000
231	Siro Bisi	50	60000
252	Nico Bini	44	70000
301	Steve Smith	34	70000
375	Mary Smith	50	65000

Dipendenze

Capo	Impiegato
210	101
210	103
210	104
231	105
301	210
301	231
375	252

- La query “fornire i numeri di matricola, nomi ed età degli impiegati che guadagnano più di 40000 Euro all’anno” può essere formulata in algebra relazionale come segue:

$$\pi_{\text{Numero,Nome,Età}}(\sigma_{\text{Stipendio}>40000}(\text{Impiegati}))$$

Esempio di query

- La query "fornire i numeri di matricola dei capi degli impiegati che guadagnano più di 40000 Euro all'anno" coinvolge entrambe le relazioni tramite un equi-join:

$$\pi_{Capo}(\text{Dipendenze} \bowtie_{\text{Impiegato=Numero}}(\sigma_{\text{Stipendio}>40000}(\text{Impiegati})))$$

- La query "fornire i nomi e gli stipendi dei capi degli impiegati che guadagnano più di 40000 Euro all'anno" richiede un ulteriore join, rispetto alla query precedente, con la relazione Impiegati in quanto nomi e stipendi non sono attributi che si trovano nella relazione Dipendenze:

$$\pi_{\text{NomeC,StipendioC}}(\rho_{\text{NumeroC,NomeC,EtàC,StipendioC} \leftarrow \text{Numero,Nome,Età,Stipendio}}(\text{Impiegati}) \bowtie_{\text{NumeroC=Capo}} (\text{Dipendenze} \bowtie_{\text{Impiegato=Numero}}(\sigma_{\text{Stipendio}>40000}(\text{Impiegati}))))$$

Si noti la necessità di rinominare gli attributi della relazione Impiegati quando si riferiscono ai capi (in modo da distinguerli dagli omonimi attributi che si riferiscono invece agli impiegati comuni).

Esempio di query

- Per scrivere in algebra relazionale la query "fornire i numeri di matricola ed i nomi dei capi i cui dipendenti guadagnano *tutti* più di 40000 Euro all'anno", occorre far uso dell'operatore insiemistico di differenza, in quanto l'espressione "tutti" presuppone una quantificazione universale e non esiste un operatore in grado di esprimerla direttamente:

$$\pi_{\text{Numero,Nome}}(\text{Impiegati} \bowtie_{\text{Numero=Capo}}$$
$$(\pi_{\text{Capo}}(\text{Dipendenze}) -$$
$$\pi_{\text{Capo}}(\text{Dipendenze} \bowtie_{\text{Impiegato=Numero}}(\sigma_{\text{Stipendio}\leq 40000}(\text{Impiegati})))))$$

- Quindi la tecnica usata è quella di selezionare dapprima i numeri di matricola dei capi con dipendenti che guadagnano 40000 o meno di 40000 Euro all'anno. Poi tali numeri vengono sottratti dall'insieme dei numeri di matricola di tutti i capi, lasciando così soltanto quelli voluti in modo da eseguire il join più esterno con la relazione Impiegati ed ottenere anche i nomi dei capi.