Algebra relazionale

RIEPILOGO E OTTIMIZZAZIONE

DOCENTE
PROF. ALBERTO BELUSSI

Anno accademico 2018/19

Riepilogo operatori algebra

- **Operatori insiemistici**

  Applicabili SOLO a relazioni con lo stesso schema:

  BASE

  - **Unione:** $r_1 \cup r_2 = \{t \mid t \in r_1 \vee t \in r_2\}$
  - **Differenza:** $r_1 - r_2 = \{t \mid t \in r_1 \wedge t \notin r_2\}$

  DERIVATI

  - **Intersezione :** $r_1 \cap r_2 = r_1 - (r_1 - r_2)$

Riepilogo operatori algebra

- **Operatori specifici**

  Applicabili a singole relazioni:

  BASE

  - **Ridenominazione:**

$$\rho_{A1\ A2\ \ldots Ak \rightarrow\ B1\ B2\ \ldots Bk}(r) =$$

$$\{t \mid \exists t' \in r:\ \forall i \in \{1,\ldots,k\}:\ t[B_i] = t'[A_i]\}$$

  - **Selezione:**

$$\sigma_F(r) = \{t \mid t \in r \land F(t)\}$$

  - **Proiezione**

$$\Pi_Y(r) = \{t \mid \exists t' \in r:\ \forall A_i \in Y:\ t[A_i] = t'[A_i]\}$$

Riepilogo operatori algebra

- **Operatori di giunzione (join)**

  Applicabili a coppie di relazioni:

  BASE

  - **Join naturale** (su r_1 di schema X_1 e r_2 di schema X_2):

$$r_1 \bowtie r_2 =$$

$$\{t \mid \exists t_1 \in r_1 : \exists t_2 \in r_2 : t_1 = t[X_1] \wedge t_2 = t[X_2] \}$$

  DERIVATI

  - **ϑ-Join** (su r_1 di schema X_1 e r_2 di schema X_2 con $X_1 \cap X_2 = \emptyset$)

$$r_1 \bowtie_F r_2 = \sigma_F (r_1 \bowtie r_2)$$

Equivalenza tra gli operatori di join

Equivalenza tra join naturale e ϑ-Join(equi-join)

Il join naturale tra due relazione r_1 di schema X_1 e r_2 di schema X_2 dove $X_1 \cap X_2 = \{c_1, \ldots, c_m\}$ equivale alla seguente espressione contenente un ϑ-Join:

$$r_1 \bowtie r_2 \equiv$$

$$\Pi_{X1 \cup X2} (r_1 \bowtie_{c1'=c1 \wedge \ldots \wedge cm'=cm} (\rho_{c1, c2, \ldots, cm \rightarrow c1', c2', \ldots, cm'} (r_2)))$$

Algebra con valori nulli

Algebra relazionale con valori nulli

E' opportuno estendere l'algebra relazionale affinché possa manipolare anche i valori nulli. Le operazioni che devono essere raffinate in presenza di valori nulli sono:

- Selezione: le condizioni di selezione in presenza di valori nulli hanno i seguenti valori di verità:
  - A ϑ B sulla tupla t: se t[A] o t[B] sono NULL allora t[A] ϑ t[B] è FALSO.
  - A ϑ cost sulla tupla t: se t[A] è NULL allora t[A] ϑ const è FALSO.
  - Condizioni atomiche aggiuntive: A is null / A is not null
- Join naturale: la condizione di uguaglianza sugli attributi comuni alle due relazioni <u>è falsa su t1 e t2 se almeno uno degli attributi comuni di t1 o t2 è NULL</u>.

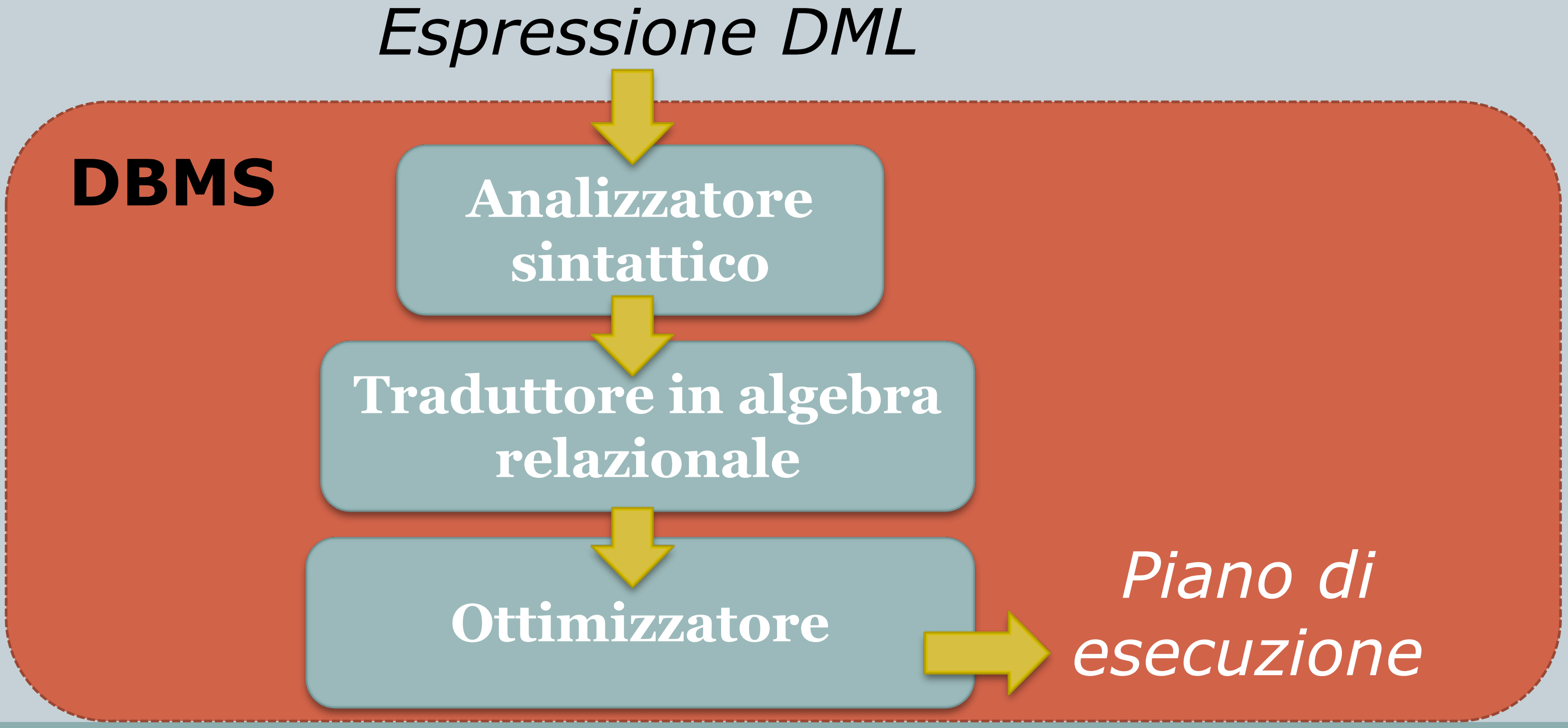
Join esterni

Consentono di ottenere nel risultato del join tutte le tuple (anche le tuple pendenti “dangling tuples”) di una o di entrambe le relazioni coinvolte nel join, eventualmente estese con valori nulli.

- LEFT JOIN $r_1 \bowtie_{LEFT} r_2$
- RIGHT JOIN $r_1 \bowtie_{RIGHT} r_2$
- FULL JOIN $r_1 \bowtie_{FULL} r_2$

Ottimizzazione di espressioni DML

Ogni espressione DML (solitamente specificata in linguaggio dichiarativo) ricevuta dal DBMS è soggetta ad un processo di elaborazione.



Ottimizzatore

L’ottimizzatore genera un’espressione equivalente all’interrogazione di input e di costo inferiore.

Il costo viene valutato in termini di **dimensione dei risultati intermedi**.

L’ottimizzatore esegue **trasformazioni di equivalenza** allo scopo di RIDURRE LA DIMENSIONE DEI RISULTATI INTERMEDI.

Equivalenza tra espressioni algebriche

- Equivalenza dipendente dallo schema: dato uno schema R

  $E1 \equiv_R E2$ se $E1(r) = E2(r)$ per ogni istanza r di schema R

- Equivalenza assoluta: è indipendente dallo schema

  $E1 \equiv E2$ se $E1 \equiv_R E2$ per ogni schema R compatibile con E1 e E2

Trasformazioni di equivalenza

Sia ***E* un'espressione di schema *X***, si definiscono le seguenti trasformazioni di equivalenza:

- <u>Atomizzazione delle selezioni</u>

$$\sigma_{F1 \wedge F2}(E) \equiv \sigma_{F1}(\sigma_{F2}(E))$$

È propedeutica ad altre trasformazioni. Non ottimizza se non è seguita da altre trasformazioni.

- <u>Idempotenza delle proiezioni</u>

$$\Pi_{Y}(E) \equiv \Pi_{Y}(\Pi_{YZ}(E)) \text{ dove } Z \subseteq X$$

È propedeutica ad altre trasformazioni. Non ottimizza se non è seguita da altre trasformazioni.

Trasformazioni di equivalenza

Siano **E1 e E2 espressioni di schema X1 e X2**, si definiscono le seguenti trasformazioni di equivalenza:

- Anticipazione delle selezioni rispetto al join:

$$\sigma_F(E1 \bowtie E2) \equiv E1 \bowtie \sigma_F(E2)$$

Applicabile solo se F si riferisce solo ad attributi di E2.

- Anticipazione della proiezione rispetto al join:

$$\Pi_{X1Y}(E1 \bowtie E2) \equiv_R E1 \bowtie \Pi_Y(E2)$$

Applicabile solo se $Y \subseteq X2$ e $(X2 - Y) \cap X1 = \emptyset$

Trasformazioni di equivalenza

Combinando l'anticipazione della proiezione con l'idempotenza delle proiezioni otteniamo:

$$\Pi_Y(E1 \bowtie_F E2) \equiv \Pi_Y(\Pi_{Y_1}(E1) \bowtie_F \Pi_{Y_2}(E2))$$

$$\Pi_Y(E1 \bowtie E2) \equiv \Pi_Y(\Pi_{Y_1}(E1) \bowtie \Pi_{Y_2}(E2))$$

dove:

- $Y1 = (X1 \cap Y) \cup J1$
- $Y2 = (X2 \cap Y) \cup J2$
- J1/2 sono gli attributi di E1/2 coinvolti nel join (vale a dire presenti in F per il theta-join, mentre in caso di join naturale $J1=J2= X1 \cap X2$)

Ulteriori Trasformazioni di equivalenza

Siano E1 e E2 espressioni di schema X1 e X2, si definiscono le seguenti trasformazioni di equivalenza:

- Inglobamento di una selezione in un prodotto cartesiano (attenzione questa regola si applica solo dopo aver verificato che non sia possibile anticipare selezioni rispetto al join):

$$\sigma_F(E1 \bowtie E2) \equiv E1 \bowtie_F E2$$

dove $X1 \cap X2 = \varnothing$.

Ulteriori Trasformazioni di equivalenza

- Applicazione delle proprietà commutativa e associativa di: unione, join naturale (o prodotto cartesiano), intersezione.
- Applicazione della proprietà distributiva:

$$\sigma_F(E1 \cup E2) \equiv \sigma_F(E1) \cup \sigma_F(E2)$$

$$\sigma_F(E1 - E2) \equiv \sigma_F(E1) - \sigma_F(E2)$$

$$\Pi_Y(E1 \cup E2) \equiv \Pi_Y(E1) \cup \Pi_Y(E2)$$

$$E1 \bowtie (E2 \cup E3) \equiv (E1 \bowtie E2) \cup (E1 \bowtie E3)$$

Ulteriori Trasformazioni di equivalenza

- Applicazione di altre trasformazioni:

$$\sigma_{F1 \vee F2}(E) \equiv \sigma_{F1}(E) \cup \sigma_{F2}(E)$$

$$\sigma_{F1 \wedge F2}(E) \equiv \sigma_{F1}(E) \cap \sigma_{F2}(E) \equiv \sigma_{F1}(E) \bowtie \sigma_{F2}(E)$$

$$\sigma_{F1 \wedge \neg F2}(E) \equiv \sigma_{F1}(E) - \sigma_{F2}(E)$$

Esempio di ottimizzazione

TRENO(<u>NumTreno</u>, Cat, OraPart, MinPart, OraArr, MinArr, Dest)

FERMATA(<u>NumTreno, Stazione</u>, Ora, Min)

Q: Trovare la destinazione dei treni non regionali che fermano a "Peschiera".

<u>Dimensioni dei dati</u>:

TRENO:
#attributi = 7
#tuple = 150
#tuple treni regionali = 40

FERMATA:
#attributi = 4
#tuple = 1000
#tuple staz peschiera = 50
#tuple staz peschiera e treno non regionale = 45

Esempi

TRENO(Num, Cat, Part, Arrivo, Dest)

FERMATA(Treno, Stazione, Orario)

1) Trovare i treni (tutti gli attributi) che partono dopo le 12.00 e prima delle 16.00 e non sono regionali.
2) Trovare tutte le fermate dei treni che partono dalla stazione di Verona P.N.
3) Trovare i treni IC per Venezia Mestre (destinazione o fermata a Venezia Mestre) riportando il numero del treno, l'orario di partenza e l'orario di arrivo (o di fermata) a Venezia Mestre.

Esempi

4) Trovare i treni regionali che fermano a Vicenza riportando il numero del treno e l’orario di partenza.
5) Trovare il nome delle stazioni dove dopo le 20.30 ferma uno e un sol treno.