12-10-2022

ALGORITMO DI QUERY OPTIMIZATION

- Due espressioni sono **equivalenti** se producono lo stesso risultato qualunque sia l'istanza attuale della base di dati
  Tutti i DB prendono la query e viene *riarrangiata* per **trasformarla** in **forma meno costosa a livello computazionale**.

  > prima di eseguire la query si crea una **versione più efficiente** equivalente

Un'equivalenza importante

- Push selection (se A è attributo di R_2)

$$\sigma_{A=10}(R_1 \bowtie R_2) = R_1 \bowtie \sigma_{A=10}(R_2)$$

- Riduce in modo significativo la dimensione del risultato intermedio (e quindi il costo dell'operazione)

In R2 le tuple sono minori

1. Raggruppamento di restrizioni

$$\sigma_{c(X)}\left(\sigma_{c(Y)}(E)\right) = \sigma_{c(X) \wedge c(Y)}(E)$$

> 2. Commutatività di σ e π
>
> a. $\sigma_{c(X)}(\pi_Y(E)) = \pi_Y\left(\sigma_{c(X)}(E)\right)$ $se\ X \subseteq Y$
>
> b. $\pi_Y\left(\sigma_{c(X)}(\pi_{XY}(E))\right) = \pi_Y\left(\sigma_{c(X)}(E)\right)$ $se\ X \nsubseteq Y$
>
> 2.b: posso fare l'unione di attributi. la proiezione interna è inutile.

```
se ci sono proiezioni inutili le tolgo perchè il risultato non cambia.
```

3. Anticipazione di σ rispetto a $\times$.

a. $\sigma_{c(X)}(E \times F) = \sigma_{c(X)}(E) \times F\ se\ X \subseteq attr(E)$

b. $\sigma_{c(X) \wedge c(Y)}(E \times F) = \sigma_{c(X)}(E) \times \sigma_{c(Y)}(F)$
$se\ X \subseteq attr(E)$ e $Y \subseteq attr(F)$

a. $\sigma_{c(X) \wedge c(Y) \wedge c(Z)}(E \times F) = \sigma_{c(Z)}\left(\sigma_{c(X)}(E) \times \sigma_{c(Y)}(F)\right)$
$se\ X \subseteq attr(E), Y \subseteq attr(F), Z \subseteq attr(E) \cup attr(F)$

> Il prodotto cartesiano equivale alla join.
>
> 4. Raggruppamento di proiezioni.
> $\pi_X(\pi_Y(E)) = \pi_X(E)\ se\ X \subseteq Y$
>
> 5. Eliminazione di proiezioni superflue.
> $\pi_X(E) = E\ se\ X = attr(E)$
>
> 6. Anticipazione della π rispetto a $\times$.
> $\pi_{XY}(E \times F) = \pi_X(E) \times \pi_Y(F)$
> $se\ X \subseteq attr(E)\ e\ Y \subseteq attr(F)$

L'algoritmo di query optimization

Utilizzando le operazioni sopra descritte possiamo ottimizzare una query con il seguente algoritmo 😊:

- si anticipano le selezioni
- raggruppamento restrizioni in modo da avere una selezione unica
  - Si applicano le seguenti tre regole (per anticipare la selezione) finché è possibile

    A. Si anticipa σ rispetto a π usando la ***2.a***
    $\sigma_{c(X)}(\pi_Y(E)) = \pi_Y\left(\sigma_{c(X)}(E)\right)$

    B. Si raggruppano le restrizioni usando la 1
    $\sigma_{c(X)}\left(\sigma_{c(Y)}(E)\right) = \sigma_{c(X)\wedge c(Y)}(E)$

    C. Si anticipa l'esecuzione di σ su $\times$ usando la 3.

Anticipazione delle proiezioni

D. Si eliminano le proiezioni superflue usando la 5
$\pi_X(E) = E \text{ se } X = attr(E)$

E. Si raggruppano le proiezioni mediante la regola 4
$\pi_X(\pi_Y(E)) = \pi_X(E) \text{ se } X \subseteq Y$

F. Si anticipa l'esecuzione delle proiezioni rispetto al prodotto usando ripetutamente la 2
[$\pi_Y\left(\sigma_{c(X)}(\pi_{XY}(E))\right) = \pi_Y\left(\sigma_{c(X)}(E)\right) \text{ se } X \not\subseteq Y$] e la 6 [Anticipazione della π rispetto a $\times$].

Esistono, inoltre altre proprietà della proiezione e, in particolare, della selezione:

Distributività

- $\sigma_C(R_1 \cup R_2) = \sigma_C(R_1) \cup \sigma_C\,(R_2)$
- $\sigma_C(R_1 - R_2) = \sigma_C(R_1) - \sigma_C\,(R_2)$
- $\pi_X(R_1 \cup R_2) = \pi_X(R_1) \cup \pi_X\,(R_2)$

- $\sigma_{C \vee D}(R) = \sigma_C(R) \cup \sigma_D\ (R)$
- $\sigma_{C \wedge D}(R) = \sigma_C(R) \cap \sigma_D\ (R)$
- $\sigma_{C \wedge \neg D}(R) = \sigma_C(R) - \sigma_D\ (R)$

Queste regole vengono effettuate quando si deve ottenere una **query efficiente**. Da queste vengono applicate tante altre ottimizzazioni.