

28 / 11 / 2024

Indicazioni sulla sintassi in presenza di operatori di confronto

Nota 1

Ripasso dalla lezione precedente.

Nota 2

In un select annidato confrontiamo due valori. (Uno proviene dal select esterno e l'altro proviene da quello interno.)

In ogni singolo passo, questi due valori:

~ devono appartenere a domini compatibili (es: un valore integer e un valore decimal);

e inoltre

~ devono possedere l'operatore di confronto usato (es: l'operatore <).

Nota 3

Nella medesima clausola di confronto (che può essere un WHERE oppure un HAVING) del select esterno, possiamo combinare due o più confronti mediante i consueti operatori logici.

Es:

```
/* Select Esterno */
..
WHERE Este1 = (
    SELECT Inte1
    ..
)
AND Este2 = (
    SELECT Inte2
    ..
)
/* Eventuale continuazione del Select esterno. */;
```

Select interno indipendente

Un select interno è **indipendente** quando usa soltanto tabelle indicate nel proprio FROM.

Quindi:

~ Il risultato del select interno non dipende dalle righe del select esterno.

~ Sql esegue il select interno soltanto una volta, prima del select esterno.

Select interno collegato

Un select interno è **collegato** quando non è indipendente, cioè quando usa almeno una tabella che non compare nel FROM del select interno. Quindi:

~ Il risultato del select interno dipende dalla particolare riga del select esterno che sta per essere confrontata con il select interno.

~ Sql esegue il select interno per ogni riga del select esterno.

Es:

Il select interno dell'esempio seguente è collegato perché usa (nella clausola `WHERE`) la tabella `T1` che non compare nel `FROM` del select interno.

```
SELECT T1.K, T1.X
FROM T1
WHERE T1.X = (
    SELECT MAX(T2.X)
    FROM T2
    WHERE T1.Y = T2.Y
);
```

Vedremo il suo effetto più avanti.

Note importanti

Riepilogo le note più importanti tra quelle che ho descritto precedentemente.

- 1 Un select interno serve a selezionare qualcosa del select esterno.
Quindi deve essere inserito nel `WHERE` o nell'`HAVING` del select esterno.
- 2 La selezione suddetta avviene mediante un operatore di confronto applicato tra:
 - ~ l'(unica) espressione che si trova nel `WHERE` [o nell'`HAVING`] del select esterno
 - e
 - ~ l'(unica) espressione che si trova nella clausola `SELECT` del select interno.
 Quindi, la clausola `SELECT` del select interno deve avere un unico argomento.
- 3 Per ogni iterazione del select esterno,
 - La riga [o il gruppo], composta da una colonna, individuata dal suo `WHERE` [o dal suo `HAVING`]
 - deve essere confrontata con
 - la tabella individuata dal select interno, composta da una colonna e da zero, una o più righe.
- 4 Ci sono due tipologie di select interno:
 - ~ Indipendente,
 - ~ Collegato.

Sintassi e risultato di ogni modalità di confronto

Ci sono tre modalità per confrontare l'espressione del select interno con quella del select esterno.

Modalità 1

Questa modalità può essere usata soltanto se il select interno individua al massimo una riga; altrimenti, Sql darà un errore in esecuzione.

Quindi, per essere certi che l'interrogazione funzioni sempre (cioè, con qualsiasi input), dovremo progettare adeguatamente il select interno.

Sintassi:

```

..
[WHERE | HAVING] EspreEste OperatoreDiConfronto (
    SELECT EspreInte
    ..
);

```

Risultato

Il risultato mostra ogni riga [o gruppo] del select esterno che soddisfa il confronto seguente:

```
EspreEste OperatoreDiConfronto EspreInte
```

Es.1

```

/* Select Esterno */
..
WHERE T1.X >= ( /* Il select interno individua sicuramente al massimo una riga
    se T2.K e` superchiave. */
    SELECT T2.X
    FROM T2
    WHERE T2.K = 5
);

```

Es.2

```

/* Select Esterno */
..
HAVING COUNT(T1.X) >= ( /* Il select interno individua sempre un'unica riga. */
    SELECT COUNT(T2.Y)
    ..
);

```

Esempio importante

Consideriamo una tabella $T1(K, X)$, dove X appartiene a un dominio che possiede l'operatore $>$. (Ad es., X è un numero.)

Dobbiamo trovare i valori di K che corrispondono al massimo di X . Questo è il cosiddetto problema della ricerca dei **punti di massimo**.

Risolviamo il problema mediante il seguente select annidato, dove il select interno è indipendente.

```

SELECT T1.K, T1.X
FROM T1
WHERE T1.X = (
    SELECT MAX(T1.X)
    FROM T1
);

```

SOLU.Aeroporti.N

Elencate i tipi di aereo aventi la massima capienza.

SOLU.Aeroporti.O

Elencate le città che sono servite dai tipi di aereo aventi la massima capienza.

L'elenco deve avere un'unica colonna.

Sintassi e risultato di ogni modalità di confronto

Modalità 2

Questa modalità può essere usata qualunque sia il numero (anche zero) di righe individuate dal select interno.

Sintassi

Tra l'espressione del select esterno e l'inizio del select interno deve dobbiamo inserire una a scelta di queste quattro forme alternative:

```
OperatoreDiConfronto ANY
OperatoreDiConfronto ALL
IN
NOT IN
```

Quindi, il select annidato si presenta in questa forma:

```
/* Select esterno */
..
[WHERE | HAVING] EspreEste UnaDelleQuattroFormeSuddette (
    SELECT EspreInte
    ..
);
```

Risultato

Una riga [o un gruppo] del select esterno compare nel risultato a seconda della parola chiave indicata:

~ con ANY:

la riga compare se *EspreEste* soddisfa il confronto con almeno un valore del select interno.

~ con ALL:

la riga compare se *EspreEste* soddisfa il confronto con ogni valore del select interno.

~ con IN:

la riga compare se *EspreEste* è tra i valori del select interno.

(Quindi, *IN* equivale a = ANY).

~ con NOT IN:

la riga compare se *EspreEste* non è tra i valori del select interno.

(Quindi, *NOT IN* equivale a <> ALL).

Es:

```
/* Select esterno */  
..  
WHERE T1.X >= ANY (  
    SELECT T2.X  
    FROM T2  
);
```