
Basi di Dati
Modulo Laboratorio

Lezione 5: Interrogazioni nidificate

DR. SARA MIGLIORINI

Interrogazioni nidificate

- Un'interrogazione è nidificata quando è presente all'interno di un'altra interrogazione.
- Un'interrogazione dentro la clausola FROM è un esempio di interrogazione nidificata.

Interrogazione dentro clausola FROM

```
SELECT titolo, prezzoIntero  
FROM Mostra, ( SELECT MAX ( prezzoIntero )  
                FROM Mostra ) AS T( prezzoMax )  
WHERE prezzoIntero = prezzoMax;
```

Interrogazioni nidificate

- SQL permette di fare un'interrogazione nidificata anche dentro la clausola **WHERE**.
- L'interrogazione nidificata può essere usata per costruire selezioni di righe della query principale più sofisticati.
- In lezioni precedenti si è introdotto che la selezione è spesso basata sul confronto tra il valore di un attributo e valori di altri attributi della stessa riga o valori costanti.
- Se si usano interrogazioni nidificate, il **confronto** è tra un valore di un attributo (**valore singolo**) e il risultato di una interrogazione (**possibile insieme di valori**).
- Quindi:
 - Gli operatori di confronto tradizionali (<,>,<=>,...) NON non possono essere usati.
 - Si devono usare dei nuovi operatori, **[NOT] EXISTS**, **[NOT] IN**, **ALL**, **ANY/SOME**, che estendono i tradizionali operatori a questo tipo di confronti.

Interrogazioni nidificate: Operatore EXISTS

Sintassi

EXISTS (subquery)

- (subquery) è una **SELECT**.
- **EXISTS** ritorna falso se (subquery) non contiene righe; vero altrimenti.
- **EXISTS** è significativo quando nella (subquery) si selezionano righe usando i valori della riga corrente nella **SELECT** principale: **data binding**

Molto costoso

Interrogazioni nidificate: Operatore EXISTS

Esempio

Determinare i nomi degli impiegati che sono diversi tra loro ma di pari lunghezza.

```
SELECT l. nome  
FROM Impiegato l  
WHERE EXISTS (  
    SELECT 1 FROM Impiegato l1 WHERE l.nome <> l1. nome  
    AND CHAR_LENGTH( l.nome ) = CHAR_LENGTH( l1. nome )  
);
```

l.nome nella subquery è il valore di nome nella riga corrente della SELECT principale.

Interrogazioni nidificate: Operatore EXISTS

Nota

Se si usano attributi esterni nella subquery (data binding), la subquery deve essere valutata per ogni riga della **SELECT** principale.

Interrogazioni nidificate: Operatore EXISTS

Esempio

Visualizzare il nome e il cognome dei docenti, escludendo i coordinatori, che hanno tenuto almeno due insegnamenti (o moduli) con più di 24 crediti nel 2010/2011.

```
SELECT P.nome , P.cognome
FROM Persona P
WHERE EXISTS (
    SELECT 1 FROM Docenza D JOIN InsErogato IE ON D. id_inserogato = IE.id
    WHERE IE. crediti > 24 AND D. coordinatore = '0'
    AND IE. annoaccademico = '2010/2011' AND D. id_persona = P.id
    GROUP BY D. id_persona
    HAVING COUNT (*) >=2
);
```

Interrogazioni nidificate: Operatore EXISTS

Nota

L'operatore **NOT** può essere usato in coppia con **EXISTS**.

Esempio

Visualizzare il nome dei corsi di studio che nel 2006/2007 non hanno erogato insegnamenti il cui nome contiene la sottostringa 'Info'.

```
SELECT CS. nome FROM CorsoStudi CS
WHERE NOT EXISTS (
    SELECT 1 FROM InsErogato IE JOIN Insegn I ON IE. id_insegn = I.id
    WHERE I.nomeins LIKE '%Info%'
    AND IE.annoaccademico = '2006/2007'
    AND IE.id_corsostudi = CS.id);
```


Interrogazioni nidificate: Operatore IN

Sintassi

```
[ROW]( expr [ ,...]) IN ( subquery )
```

- **expr** è un'espressione costruita con un attributo della query principale. Ci possono essere una o più espressioni.
- La (**subquery**) deve restituire un numero di colonne pari al numero di espressioni in (**expr** [...]).
- I valori dell'espressioni vengono confrontati con i valori di ciascuna riga del risultato di (subquery).
- Il confronto ritorna vero se i valori sono uguali ai valori di almeno una riga della subquery.

Interrogazioni nidificate: Operatore IN

Esempio

```
SELECT l.nome , l.cognome  
FROM Impiegato l  
WHERE ROW( l.nome , l.cognome ) IN (  
    SELECT l1.nome, l1. cognome FROM ImpiegatoAltraAzienda l1  
);
```

Interrogazioni nidificate: Operatore ANY/SOME

Sintassi

expression operator ANY(subquery)
expression operator SOME(subquery)

- (subquery) è una SELECT che deve restituire UNA sola colonna;
- expression è un'espressione che coinvolge attributi della SELECT principale.
- operator è un operatore di confronto, come '=', '>=', ...
- ANY: ritorna vero se expression è operator rispetto al valore di una qualsiasi riga del risultato di (subquery).
- SOME è uno sinonimo di ANY.

Interrogazioni nidificate: Operatore ANY/SOME

Esempio

Visualizzare il nome degli insegnamenti che hanno un numero di crediti inferiore alla media dell'ateneo di un qualsiasi anno accademico.

```
SELECT DISTINCT I.nomeins , IE.crediti
FROM Insegn I JOIN InsErogato IE ON I.id=IE.id_insegn
WHERE IE.crediti < ANY (
    SELECT AVG ( crediti ) FROM InsErogato
    WHERE modulo =0
    GROUP BY annoaccademico
);
```

11607 righe

Interrogazioni nidificate: Operatore ALL

Sintassi

expression operator ALL(subquery)

- (subquery) è una SELECT che deve restituire UNA sola colonna;
- expression è un'espressione che coinvolge attributi della SELECT principale.
- operator è un operatore di confronto, come '=', '>=', ...
- ALL: ritorna vero se expression è operator rispetto al valore di ciascuna riga del risultato di (subquery).

Interrogazioni nidificate: Operatore ALL

Esempio

Trovare il nome degli insegnamenti con almeno un docente e crediti maggiori rispetto ai crediti di ciascun insegnamento del corso di laurea con id=6. (si considerano solo occorrenze insegnamento genitore).

```
SELECT DISTINCT I. nomeins , IE. crediti
FROM Insegn I JOIN InsErogato IE ON I.id = IE. id_insegn
      JOIN Docenza D ON IE.id = D. id_inserogato
WHERE IE. modulo = 0 AND IE. crediti > ALL (
  SELECT crediti FROM InsErogato
  WHERE id_corsostudi = 6 AND modulo =0
);
```

(1485 righe)

Operatore ALL: esempio complicato

- Quesito: Trovare il nome degli insegnamenti con almeno due docenti e con crediti maggiori rispetto ai crediti di ciascun insegnamento del corso di laurea con id=6. (si considerano solo occorrenze insegnamento genitore).
- Possibile soluzione: usare due volte la soluzione che trova i dati per almeno un docente assicurando che il nome dell'insegnamento sia lo stesso e che i docenti siano diversi.

Operatore ALL: esempio complicato

Esempio

```
SELECT DISTINCT I.nomeins
FROM Insegn I JOIN InsErogato IE1 ON I.id = IE1.id_insegn
      JOIN Docenza D1 ON IE1.id = D1.id_inserogato
      JOIN InsErogato IE2 ON I.id = IE2.id_insegn
      JOIN Docenza D2 ON IE2.id = D2.id_inserogato
WHERE IE1.modulo = 0 AND IE1.credits > ALL (
  SELECT credits FROM InsErogato
  WHERE id_corsostudi = 6 AND modulo = 0
)
AND IE2.modulo = 0 AND IE2.credits > ALL (
  SELECT credits FROM InsErogato
  WHERE id_corsostudi = 6 AND modulo = 0
)
AND D1.id_persona <> D2.id_persona;
(448 righe )
```

Il nome dell'insegnamento è il JOIN tra le due tabelle di InsErogato e ciascuna di queste è in JOIN con una alias della tabella Docente.

Operatore ALL: esempio complicato (ALT.)

Esempio (Soluzione alternativa)

Selezioniamo le coppie (nomeins, docente) con le caratteristiche volute circa i crediti degli insegnamenti erogati associati.

```
SELECT DISTINCT I.nomeins , D.id_persona
FROM Insegn I
  JOIN InsErogato IE ON I.id = IE.id_insegn
  JOIN Docenza D ON IE.id = D.id_inserogato
WHERE IE. modulo = 0 AND IE.crediti > ALL (
  SELECT crediti FROM InsErogato
  WHERE id_corsostudi = 6 AND modulo =0
);
```

(2277 righe)

Questa query è il punto di partenza: raggruppando il risultato della query per nomeins, si hanno gruppi dove ciascuno contiene tante righe quanti sono i docenti che hanno insegnato quel corso. Basta quindi imporre un vincolo sulla numerosità di ciascun gruppo per ottenere il risultato.

Operatore ALL: esempio complicato (ALT.)

Esempio (Soluzione alternativa)

```
SELECT nomeInsegnamento
FROM (
  SELECT DISTINCT I.nomeins, D.id_persona
  FROM Insegn I JOIN InsErogato IE ON I.id = IE.id_insegn
                JOIN Docenza D ON IE.id = D.id_inserogato
  WHERE IE. modulo = 0 AND IE. crediti > ALL (
    SELECT crediti FROM InsErogato
    WHERE id_corsostudi = 6 AND modulo = 0
  )
) AS Risultato( nomeInsegnamento, persona )
GROUP BY nomeInsegnamento
HAVING COUNT(*) >= 2;
```

(448 righe)