

SQL: operazioni sui dati SELECT

Fallucchi Francesca

fallucchi@info.uniroma2.it

<http://art.uniroma2.it/fallucchi/>

22/10/2011

Interrogazioni nidificate

- Interrogazioni semplici
nella clausola "where" delle SELECT si hanno al più condizioni composte ottenute con i predicati logic.
i predicati rappresentano confronti tra due valori
- Interrogazioni Nidificate
Confronto fra uno o più attributi ed il risultato di una sottointerrogazione (query *annidata*)
Di solito prima si esegue la query piu' interna, ma ci sono eccezioni!!!

Interrogazioni nidificate

- Sintassi

```
SELECT { * | <val> [, <val> ... ] }  
FROM <tableref> [, <tableref> ... ]  
[ WHERE <search_condition> ]
```

dove

```
<search_condition> = <val> <operator>  
{ <val> | ( <select_one> ) }  
| <val> [ NOT ] IN ( <val> [, <val> ... ] ) |  
( <select_list> )  
| { ALL | SOME | ANY } ( <select_list> )  
| .....
```

Esempio

- Vogliamo selezionare il nome e il reddito del padre di Franco

```
SELECT Nome, Reddito  
FROM Persone, Paternita  
WHERE Nome = Padre and Figlio = 'Franco'
```



```
SELECT Nome, Reddito  
FROM Persone  
WHERE Nome = (SELECT Padre  
FROM Paternita  
WHERE Figlio = 'Franco')
```

Interrogazioni nidificate, commenti

- La forma nidificata è "meno dichiarativa", ma talvolta più leggibile
- Solitamente richiede meno variabili
- Le sottointerrogazioni non possono contenere operatori insiemistici
"l'unione si fa solo al livello esterno"
la limitazione non è significativa
- Per evitare disomogeneità dei termini di confronto si estendono gli operatori con:
all
any

Esempio

- Tutti i dati degli impiegati che lavorano in dipartimento in Firenze

```
select *  
from Impiegato  
where Dipart = any (select Nome  
from Dipartimento  
where Citta = 'Firenze')
```

Esempio

- Impiegati che hanno lo stesso nome di impiegati del dip di Produzione

```
SELECT I1.Nome
FROM Impiegato I1, Impiegato I2
WHERE I1.Nome = I2.Nome and I2.Dipart = 'Produzione'
```

```
SELECT Nome
FROM Impiegato
WHERE Nome = any (select Nome
from Impiegato
where Dipart = 'Produzione')
```

Esempio

-

```
SELECT Nome
FROM Dipartimento
WHERE Nome <> ALL (SELECT Dipart
FROM Impiegato
WHERE Cognome = 'Rossi')
```

NOTA: <> ALL corrisponde a **not in**

Esempio

- Dipartimento dell'impiegato che guadagna lo stip massimo

```
SELECT Dipart
FROM Impiegato
WHERE Stipendio = (SELECT max(Stipendio)
FROM Impiegato)
```

```
SELECT Dipart
FROM Impiegato
WHERE Stipendio >= all (SELECT Stipendio
FROM Impiegato)
```

Esempio: passaggio di binding

- Data: Persona(CodFiscale, Nome, Cognome, Citta), trovare persone che hanno omonimi

```
SELECT *
FROM Persona P
WHERE exist (SELECT *
FROM Persona P1
WHERE P1.Nome=P.Nome AND
P1.Cognome=P.Cognome AND
P1.CodFiscale<>P.CodFiscale)
```

Esempio: passaggio di binding

- Data: Persona(CodFiscale, Nome, Cognome, Citta), trovare persone che **non** hanno omonimi

```
SELECT *
FROM Persona P
WHERE not exist (SELECT *
FROM Persona P1
WHERE P1.Nome=P.Nome AND
P1.Cognome=P.Cognome AND
P1.CodFiscale<>P.CodFiscale)
```

Altra Soluzione

- Data: Persona(CodFiscale, Nome, Cognome, Citta), trovare persone che **non** hanno omonimi

```
SELECT *
FROM Persona P
WHERE (Nome,Cognome) <> all
(SELECT P1.Nome, P1.Cognome
FROM Persona P1
WHERE P1.Nome=P.Nome AND
P1.Cognome=P.Cognome AND
P1.CodFiscale<>P.CodFiscale)
```

Interrogazioni nidificate

- regole di visibilità:
 - non è possibile fare riferimenti a variabili definite in blocchi più interni
 - in un blocco si può fare riferimento a variabili definite in blocchi più esterni

Esempio: regole di visibilità

```
SELECT *
FROM Impiegato
WHERE Dipart IN
    (SELECT Nome
     FROM Dipartimento D1
     WHERE Nome = 'Produzione') OR
Dipart IN
    (SELECT Nome
     FROM Dipartimento D2
     WHERE D1.CITTA' = D2.CITTA')
```

 Non è visibile