SQL – seconda parte

DOCENTE
PROF. ALBERTO BELUSSI

Anno accademico 2018/19

Interrogazioni nidificate

Interrogazioni nidificate

Si ottiene una interrogazione nidificata quando nella clausola WHERE compare un predicato complesso, vale a dire un predicato che contiene un'altra interrogazione SQL:

SELECT …
FROM …
WHERE <espr> θ' (SELECT … FROM … WHERE)

PREDICATO COMPLESSO

Interrogazioni nidificate

Predicato complesso (o a struttura complessa)

E' un predicato che confronta:

- il valore di un attributo (o un'espressione) con
- il risultato di un'altra interrogazione SQL (interrogazione nidificata).
  ATTENZIONE: l'interrogazione nidificata nel caso tipico è mono-attributo (un solo attributo nella clausola SELECT) e produce un insieme di valori.

Poiché il risultato di una interrogazione SQL è un insieme di valori occorre estendere gli operatori di confronto per poter realizzare la comparazione tra un valore e un insieme di valori.

WHERE <espr> θ' (SELECT ... FROM ... WHERE)

VALORE **INSIEME di VALORI**

Predicato complesso

Operatori di confronto per predicati complessi

Si ottengono combinando i normali operatori di confronto op ∈ {=, <>, <, >, <=, >=} con le parole chiave ALL e ANY. Il significato è il seguente:

- *A op **ANY** (SQLquery)*
  questo predicato è soddisfatto dalla tupla *t* se esiste almeno un valore *v contenuto* nel risultato dell'interrogazione *SQLquery* che verifica la condizione:

  $$t[A] \; op \; v$$

- *A op **ALL** (SQLquery)*
  questo predicato è soddisfatto dalla tupla *t* se per ogni valore *v* contenuto nel risultato dell'interrogazione *SQLquery* è verificata la condizione:

  $$t[A] \; op \; v$$

Predicato complesso

Operatori di confronto per predicati complessi

Stenografie:

- ***=ANY si può scrivere IN***
- ***<>ALL si può scrivere NOT IN***

Interrogazioni nidificate: esempi

Esempio

Trovare la destinazione dei treni che non fermano a Brescia.

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

FERMATA(NumTreno, Stazione, Orario)

```
SELECT DISTINCT Dest
FROM TRENO
WHERE NumTreno NOT IN
    (SELECT NumTreno FROM FERMATA
     WHERE Stazione = 'Brescia')
```

Interrogazioni nidificate: classificazione

Interrogazioni nidificate

Si possono classificare in due categorie:

- **Interrogazioni nidificate INDIPENDENTI dall'interrogazione che le contiene (*interrogazione esterna*)**: in questo caso l'interrogazione nidificata può essere valutata una volta sola in quanto non dipende dalla tupla corrente dell'interrogazione esterna. L'indipendenza consiste nel fatto che non ci sono variabili tupla condivise tra l'interrogazione interna e quella esterna.
- **Interrogazioni nidificate DIPENDENTI dall'interrogazione che le contiene (*interrogazione esterna*)**: in questo caso l'interrogazione nidificata condivide con l'interrogazione esterna almeno una variabile tupla che realizza il cosiddetto "*passaggio di binding*"; tale situazione implica che l'interrogazione nidificata debba essere valutata per ogni tupla dell'interrogazione esterna.

Interrogazioni nidificate: esempi

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

FILIALE(Codice, Nome, Indirizzo, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

MOVIMENTO(FilialeCC, NumeroCC, Num, Tipo, Data, Imp)

Esempio

Trovare il nome e il cognome degli intestatari dei conti dove tutti i movimenti eseguiti sono stati di importo inferiore a 1000 euro.

```
SELECT Nome, Cognome
FROM CLIENTE as C, INTESTAZIONE as I
WHERE C.CF = I.Cliente AND
      1000 >ALL (SELECT Imp FROM MOVIMENTO WHERE
      FilialeCC=I.FilialeCC AND NumeroCC=I.NumeroCC)
```

Clausola EXISTS

E' una clausola utilizzabile nei predicati complessi.

Sintassi

EXISTS (SQLquery)

Semantica

EXISTS(q)

- **true se q produce almeno una tupla**
- **false altrimenti**

EXISTS è efficace se viene applicata con passaggio di binding, vale a dire se q è una interrogazione dipendente dall'interrogazione esterna.

Clausola EXISTS

Esempio

Trovare il nome e il cognome degli intestatari di conti correnti sui quali non sono stati eseguiti prelievi BANCOMAT dal 1/4/2010 a oggi

```
SELECT Nome, Cognome
FROM CLIENTE C, INTESTAZIONE I
WHERE C.CF = I.Cliente AND NOT EXISTS
   (SELECT 1 FROM MOVIMENTO
     WHERE Data > '1/4/2010' AND
     TIPO = 'bancomat' AND FilialeCC = I.FilialeCC
     AND NumeroCC = I.NumeroCC)
```

CLAUSOLA IN E NOT IN (variante)

Utilizzando l'operatore tupla `(A1,A2, …)` (oppure `ROW(A1, A2, …)`) è possibile, nell'interrogazione nidificata che usa IN oppure NOT IN, confrontare una tupla con un insieme di tuple e non solo un valore con un insieme di valori.

Quindi per l'esempio del lucido precedente è possibile anche la seguente soluzione:

```
SELECT Nome, Cognome
FROM CLIENTE C, INTESTAZIONE I
WHERE C.CF = I.Cliente AND (I.FilialeCC, I.NumeroCC) NOT IN
   (SELECT FilialeCC, NumeroCC  FROM MOVIMENTO
    WHERE Data > '1/4/2010' AND
    TIPO='bancomat')
```

Interrogazioni nidificate

Esercizi

- Trovare il nome e il cognome dei clienti che sono intestatari di un conto insieme ad un altro cliente di cognome “Rossi”.
- Trovare numero, filiale e saldo dei conti che non hanno intestatari residenti a Verona.
- Trovare per ogni filiale il nome e il cognome del cliente correntista più giovane, riportando anche il codice della filiale.

Interrogazioni nidificate

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

- Trovare i clienti che sono intestatari di un conto insieme ad un altro cliente di cognome “Rossi”.

```
SELECT CL.Nome, CL.Cognome
FROM CLIENTE CL, INTESTAZIONE I
WHERE CL.CF = I.Cliente AND
EXISTS (SELECT 1 FROM CLIENTE C1, INTESTAZIONE I1
            WHERE C1.CF = I1.Cliente AND C1.Cognome = ‘Rossi’ AND
    I1.FilialeCC = I.FilialeCC AND I1.NumeroCC = I.NumeroCC
    AND I1.Cliente <> I.Cliente)
```