

SQL - Structured Query Language

Lab 03

Alessandro Lori

Università di Pisa

8 Aprile 2011



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX(Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX(Attributo2)
```



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX( Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX( Attributo2)
```

1. Si sceglie le tabelle da utilizzare per produrre il risultato
FROM



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX( Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX( Attributo2)
```

1. Si sceglie le tabelle da utilizzare per produrre il risultato
FROM
2. Si decide quali tuple eliminare **WHERE**



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX( Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX( Attributo2)
```

1. Si sceglie le tabelle da utilizzare per produrre il risultato **FROM**
2. Si decide quali tuple eliminare **WHERE**
3. Si raggruppano le tuple rimaste **GROUP BY**



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX( Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX( Attributo2)
```

1. Si sceglie le tabelle da utilizzare per produrre il risultato
FROM
2. Si decide quali tuple eliminare **WHERE**
3. Si raggruppano le tuple rimaste **GROUP BY**
4. Si applica l'operatore aggregato a ciascun gruppo



Riepilogo sugli operatori aggregati

Sintassi

```
SELECT Attributo1 , MAX( Attributo2) , ...  
FROM Tabella1 , Tabella2 , ...  
WHERE condizione  
GROUP BY Attributo1  
HAVING condizione su MAX( Attributo2)
```

1. Si sceglie le tabelle da utilizzare per produrre il risultato
FROM
2. Si decide quali tuple eliminare **WHERE**
3. Si raggruppano le tuple rimaste **GROUP BY**
4. Si applica l'operatore aggregato a ciascun gruppo
5. Si eliminano i gruppi che non verificano la clausola
HAVING



Esercizio

Trovare l'elenco delle madri che hanno più di un figlio



Esercizio

Trovare l'elenco delle madri che hanno più di un figlio

Soluzione

```
SELECT Madre  
FROM Maternita  
GROUP BY Madre  
HAVING COUNT( Figlio)>1
```



Esercizio

Trovare l'elenco delle persone di cui non si sa chi è la madre



Esercizio

Trovare l'elenco delle persone di cui non si sa chi è la madre

Soluzione

```
SELECT Nome  
FROM Persone  
LEFT JOIN Maternita  
ON Figlio = Nome  
WHERE Madre IS NULL
```



Interrogazioni nidificate

- ▶ Nella clausola **WHERE** si può anche inserire un confronto fra un attributo e il risultato di una sotto-interrogazione
- ▶ Naturalmente devono essere rispettati i controlli di tipo:
 - ▶ l'interrogazione nidificata deve avere un solo campo
 - ▶ il tipo del campo deve essere lo stesso

Esempio: Trovare nome e reddito del padre di Franco



Interrogazioni nidificate

- ▶ Nella clausola **WHERE** si può anche inserire un confronto fra un attributo e il risultato di una sotto-interrogazione
- ▶ Naturalmente devono essere rispettati i controlli di tipo:
 - ▶ l'interrogazione nidificata deve avere un solo campo
 - ▶ il tipo del campo deve essere lo stesso

Esempio: Trovare nome e reddito del padre di Franco

Senza query annidate

```
SELECT Nome, Reddito  
FROM Persone, Paternita  
WHERE Nome=Padre AND Figlio="Franco"
```



Interrogazioni nidificate - (2)

Con query annidate

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

- ▶ **Nota bene** Il confronto si può fare perché
 - ▶ Padre ha lo stesso tipo di Nome
 - ▶ La query più interna restituisce una sola tupla



Sottoquery con più di un risultato

SQL permette di utilizzare anche il risultato di query che restituiscono più di una tupla grazie ai seguenti costrutti:

Appartenenza (Expr **IN** (**SELECT** ...)) restituisce vero se l'espressione Expr appartiene al risultato della sottoquery



Sottoquery con più di un risultato

SQL permette di utilizzare anche il risultato di query che restituiscono più di una tupla grazie ai seguenti costrutti:

Appartenenza (Expr **IN** (**SELECT** ...)) restituisce vero se l'espressione Expr appartiene al risultato della sottoquery

Non appartenenza (Expr **NOT IN** (**SELECT** ...)) restituisce vero se l'espressione Expr non appartiene al risultato della sottoquery



Sottoquery con più di un risultato

SQL permette di utilizzare anche il risultato di query che restituiscono più di una tupla grazie ai seguenti costrutti:

Appartenenza (Expr **IN** (**SELECT** ...)) restituisce vero se l'espressione Expr appartiene al risultato della sottoquery

Non appartenenza (Expr **NOT IN** (**SELECT** ...)) restituisce vero se l'espressione Expr non appartiene al risultato della sottoquery

Esistenza (**EXISTS** (**SELECT** ...)) restituisce vero se la sottoquery ha un risultato non vuoto



Sottoquery con più di un risultato

SQL permette di utilizzare anche il risultato di query che restituiscono più di una tupla grazie ai seguenti costrutti:

Appartenenza (Expr **IN** (**SELECT** ...)) restituisce vero se l'espressione Expr appartiene al risultato della sottoquery

Non appartenenza (Expr **NOT IN** (**SELECT** ...)) restituisce vero se l'espressione Expr non appartiene al risultato della sottoquery

Esistenza (**EXISTS** (**SELECT** ...)) restituisce vero se la sottoquery ha un risultato non vuoto

Non Esistenza (**NOT EXISTS** (**SELECT** ...)) restituisce vero se la sottoquery ha un risultato vuoto



Esempi con **IN** e **EXISTS**

Esempio: Trovare nome e reddito delle persone di cui si conosce la madre o il padre



Esempi con **IN** e **EXISTS**

Esempio: Trovare nome e reddito delle persone di cui si conosce la madre o il padre

```
SELECT Nome, Reddito  
FROM Persone  
WHERE Nome IN (SELECT Figlio FROM Paternita  
                UNION  
                SELECT Figlio FROM Maternita)
```

Esempio: Trovare nome e reddito delle persone che non sono padri



Esempi con IN e EXISTS

Esempio: Trovare nome e reddito delle persone di cui si conosce la madre o il padre

```
SELECT Nome, Reddito
FROM Persone
WHERE Nome IN (SELECT Figlio FROM Paternita
               UNION
               SELECT Figlio FROM Maternita)
```

Esempio: Trovare nome e reddito delle persone che non sono padri

```
SELECT Nome, Reddito
FROM Persone
WHERE NOT EXISTS
  (SELECT Padre FROM Paternita WHERE Padre=Nome)
```



Regole di visibilità

- ▶ L'interrogazione più interna viene eseguita una volta soltanto. Il risultato viene memorizzato in una tabella temporanea e viene utilizzato per eseguire l'interrogazione più esterna.



Regole di visibilità

- ▶ L'interrogazione più interna viene eseguita una volta soltanto. Il risultato viene memorizzato in una tabella temporanea e viene utilizzato per eseguire l'interrogazione più esterna.
- ▶ Regole di visibilità



Regole di visibilità

- ▶ L'interrogazione più interna viene eseguita una volta soltanto. Il risultato viene memorizzato in una tabella temporanea e viene utilizzato per eseguire l'interrogazione più esterna.
- ▶ Regole di visibilità
 - ▶ Non è possibile utilizzare tabelle (variabili) definite in blocchi più interni (o allo stesso livello)



Regole di visibilità

- ▶ L'interrogazione più interna viene eseguita una volta soltanto. Il risultato viene memorizzato in una tabella temporanea e viene utilizzato per eseguire l'interrogazione più esterna.
- ▶ Regole di visibilità
 - ▶ Non è possibile utilizzare tabelle (variabili) definite in blocchi più interni (o allo stesso livello)
 - ▶ Si possono utilizzare tabelle definite in query più esterne



ALL

- ▶ **ALL** deve seguire un operatore di confronto e restituisce vero se la condizione espressa dall'operatore di confronto è verificata da tutte le righe della sottoquery
- ▶ **NOT IN** è semanticamente equivalente a $<>$ **ALL**
- ▶ **Esempio** Trovare Nome e Reddito delle persone che guadagnano di più



ALL

- ▶ **ALL** deve seguire un operatore di confronto e restituisce vero se la condizione espressa dall'operatore di confronto è verificata da tutte le righe della sottoquery
- ▶ **NOT IN** è semanticamente equivalente a $<>$ **ALL**
- ▶ **Esempio** Trovare Nome e Reddito delle persone che guadagnano di più

Soluzione

```
SELECT Nome, P.Reddito FROM Persone P
WHERE Reddito >= ALL (
    SELECT P1.Reddito FROM Persone P1)
```



ANY, SOME

- ▶ **ANY** deve seguire un operatore di confronto e restituisce vero se la condizione espressa dall'operatore di confronto è verificata da almeno una delle righe della sottoquery
- ▶ **IN** è semanticamente equivalente a = **ANY**
- ▶ **SOME** è un alias di **ANY**
- ▶ **Esempio** Trovare Nome e Reddito delle persone che riescono a guadagnare più di qualcun altro



ANY, SOME

- ▶ **ANY** deve seguire un operatore di confronto e restituisce vero se la condizione espressa dall'operatore di confronto è verificata da almeno una delle righe della sottoquery
- ▶ **IN** è semanticamente equivalente a = **ANY**
- ▶ **SOME** è un alias di **ANY**
- ▶ **Esempio** Trovare Nome e Reddito delle persone che riescono a guadagnare più di qualcun altro

Soluzione

```
SELECT Nome, P.Reddito FROM Persone P  
WHERE Reddito > ANY  
      (SELECT P1.Reddito FROM Persone P1)
```



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.

Senza query annidate

```
SELECT DISTINCT Pa.Padre , P.Reddito  
FROM Persone P , Paternita Pa , Persone F  
WHERE Pa.Padre = P.Nome  
AND Pa.Figlio = F.Nome  
AND F.Reddito > 20
```



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.

Con una query annidata

```
SELECT Nome, Reddito
FROM Persone
WHERE Nome IN (SELECT Padre
                  FROM Paternita , Persone
                  WHERE Figlio = Nome
                  AND Reddito > 20)
```



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.



Esercizio

Creare una query SQL che produca come risultato il nome e il reddito dei padri di persone che guadagnano più di 20 milioni.

Con due query annidate

```
SELECT Nome, Reddito
FROM Persone
WHERE Nome IN (SELECT Padre
                  FROM Paternita
                  WHERE Figlio IN (SELECT Nome
                                    FROM Persone
                                    WHERE Reddito > 20)
```



Esercizio

Creare una query SQL che restituisca il nome e il reddito dei padri di persone che guadagnano più di 20 milioni, con indicazione del reddito del figlio.



Esercizio

Creare una query SQL che restituisca il nome e il reddito dei padri di persone che guadagnano più di 20 milioni, con indicazione del reddito del figlio.

Senza query annidate

```
SELECT DISTINCT P.Nome, P.Reddito , F.Reddito  
FROM Persone AS P, Paternita , Persone AS F  
WHERE P.Nome = Padre  
AND Figlio = F.Nome  
AND F.Reddito > 20
```



Esercizio

Creare una query SQL che restituisca il nome e il reddito dei padri di persone che guadagnano più di 20 milioni, con indicazione del reddito del figlio.



Esercizio

Creare una query SQL che restituisca il nome e il reddito dei padri di persone che guadagnano più di 20 milioni, con indicazione del reddito del figlio.

Con due query annidate

```
SELECT P.Nome, P.Reddito , F.Reddito
FROM Persone AS P, Persone AS F, Paternita
WHERE Padre=P.Nome AND Figlio=F.Nome AND
P.Nome IN ( SELECT Padre
            FROM Paternita
            WHERE Figlio = ANY ( SELECT Nome
                                FROM Persone
                                WHERE Reddito > 20))
```



Esercizio

Creare una query SQL che restituisca i padri i cui figli guadagnano tutti più di venti milioni



Esercizio

Creare una query SQL che restituisca i padri i cui figli guadagnano tutti più di venti milioni

Soluzione

```
SELECT DISTINCT Padre
FROM Paternita AS Z
WHERE NOT EXISTS (SELECT *
                   FROM Paternita AS W, Persone
                   WHERE W.Padre = Z.Padre
                   AND W.Figlio = Nome
                   AND Reddito <= 20)
```

