SQL - Structured Query Language

Lab 03

Alessandro Lori

Università di Pisa

13 Aprile 2012

Creare gruppi - la clausola `GROUP BY`

- Finora utilizzando gli operatori aggregati il risultato è un unico valore
- Si possono raggruppare insiemi di tuple su cui applicare singolarmente gli operatori aggregati
- Indicando **`GROUP BY`** listaAttributi la tabella viene partizionata in gruppi
- Tutte le tuple che appartengono allo stesso gruppo hanno lo stesso valore sulla lista degli attributi specificata nel **`GROUP BY`**

Esempio

Costruire una query che produca come risultato il numero di figli di ciascun padre

```
SELECT Padre, COUNT(*)
FROM Paternita
GROUP BY Padre
```

Esempio

Costruire una query che produca come risultato il numero di figli di ciascun padre

```
SELECT Padre, COUNT(*)
FROM Paternita
GROUP BY Padre
```

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

Esempio

Costruire una query che produca come risultato il numero di figli di ciascun padre

```
SELECT Padre, COUNT(*)
FROM Paternita
GROUP BY Padre
```

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

Esempio

Costruire una query che produca come risultato il numero di figli di ciascun padre

```
SELECT Padre, COUNT(*)
FROM Paternita
GROUP BY Padre
```

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**

Esempio

Costruire una query che produca come risultato il numero di figli di ciascun padre

```
SELECT Padre, COUNT(*)
FROM Paternita
GROUP BY Padre
```

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

Attributi nel `SELECT` e `GROUP BY`

Query scorretta

```
SELECT Padre, AVG(F.Reddito), P.Reddito
FROM (Persone AS F JOIN Paternita ON Figlio = Nome)
     JOIN Persone AS P ON Padre=P.Nome
GROUP BY Padre
```

Attributi nel SELECT e GROUP BY

Query scorretta

```
SELECT Padre, AVG(F.Reddito), P.Reddito
FROM (Persone AS F JOIN Paternita ON Figlio = Nome)
     JOIN Persone AS P ON Padre=P.Nome
GROUP BY Padre
```

Query corretta

```
SELECT Padre, AVG(F.Reddito), P.Reddito
FROM (Persone AS F JOIN Paternita ON Figlio = Nome)
     JOIN Persone AS P ON Padre=P.Nome
GROUP BY Padre, P.Reddito
```

Filtrare i gruppi

Con la clausola **`HAVING`** possiamo inserire solo alcuni dei gruppi creati

Filtrare i gruppi

Con la clausola **HAVING** possiamo inserire solo alcuni dei gruppi creati

Esempio

```
SELECT Padre, AVG(F.Reddito), P.Reddito
FROM (Persone AS F JOIN Paternita ON Figlio = Nome)
     JOIN Persone AS P ON Padre=P.Nome
GROUP BY Padre, P.Reddito
HAVING AVG(F.Reddito) < 20
```

Esercizio

Costruire una query che produca come risultato i padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25 e il reddito medio di tali figli

Esercizio

Costruire una query che produca come risultato i padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25 e il reddito medio di tali figli

Soluzione

```
SELECT Padre, AVG(Reddito)
FROM Persone JOIN Paternita ON Figlio = Nome
WHERE Eta < 30
GROUP BY Padre
HAVING AVG (Reddito) > 25
```

Operatori insiemistici

- Il risultato di una **SELECT** è una tabella (insieme)
- Fra i risultati di due query si possono operare le comune operazioni sugli insiemi:
  - $\cup$ Unione - **UNION**
  - $\cap$ Intersezione - **INTERSECT**
  - $\setminus$ Complemento - **EXCEPT**

Sintassi

```
SELECT .... FROM ....
UNION / INTERSECT / EXCEPT [ALL]
SELECT .... FROM ....
```

Di norma i duplicati sono rimossi, a meno che non si aggiunga la parola chiave **ALL**

Operatori insiemistici - Requisiti

- Non è necessario che gli schemi a cui si applicano gli operatori insiemistici siano identici
- Tuttavia, gli schemi devono avere lo stesso numero di attributi e lo stesso tipo, così come specificato nelle clausole **`SELECT`**
- Nel risultato, si utilizzano come nomi di attributo quelli del primo schema

Notazione posizionale

Nel risultato di una query, gli attributi compaiono nel risultato secondo l'ordine specificato nella **`SELECT`**

Esempio - Elencare tutte le relazioni Genitore - Figlio

Notazione posizionale

Nel risultato di una query, gli attributi compaiono nel risultato secondo l'ordine specificato nella **SELECT**
Esempio - Elencare tutte le relazioni Genitore - Figlio

Soluzione errata

```
SELECT Padre, Figlio FROM Paternita
UNION
SELECT Figlio, Madre FROM Maternita
```

Notazione posizionale

Nel risultato di una query, gli attributi compaiono nel risultato secondo l'ordine specificato nella **SELECT**
Esempio - Elencare tutte le relazioni Genitore - Figlio

Soluzione errata

```
SELECT Padre, Figlio FROM Paternita
UNION
SELECT Figlio, Madre FROM Maternita
```

Soluzione corretta

```
SELECT Padre, Figlio FROM Paternita
UNION
SELECT Madre, Figlio FROM Maternita
```

Ridenominazioni

Ridenominare non cambia le cose **Esempio** - Elencare tutte le relazioni Genitore - Figlio

Ridenominazioni

Ridenominare non cambia le cose **Esempio** - Elencare tutte le relazioni Genitore - Figlio

Soluzione errata

```
SELECT Padre AS Genitore, Figlio FROM Paternita
UNION
SELECT Figlio, Madre AS Genitore FROM Maternita
```

Ridenominazioni

Ridenominare non cambia le cose **Esempio** - Elencare tutte le relazioni Genitore - Figlio

Soluzione errata

```
SELECT Padre AS Genitore, Figlio FROM Paternita
UNION
SELECT Figlio, Madre AS Genitore FROM Maternita
```

Soluzione corretta

```
SELECT Padre AS Genitore, Figlio FROM Paternita
UNION
SELECT Madre, Figlio FROM Maternita
```

Esercizio

Trovare tutte le persone che sono genitori

Esercizio

Trovare tutte le persone che sono genitori

Soluzione

```
SELECT Padre AS Genitore FROM Paternita
UNION
SELECT Madre FROM Maternita
```

Altri operatori insiemistici - **EXCEPT**

Complemento - Esempio

```
SELECT Figlio FROM Paternita
EXCEPT
SELECT Figlio FROM Maternita
```

- Restituisce le persone di cui si sa chi è il padre ma non chi è la madre
- **EXCEPT** restituisce tutti i risultati della prima query che non appartengono alla seconda query
- Non supportato da `MySQL`
- Si potrà emulare utilizzando le query annidate

Altri operatori insiemistici - INTERSECT

Intersezione - Esempio

```
SELECT Figlio FROM Paternita
INTERSECT
SELECT Figlio FROM Maternita
```

- Restituisce le persone di cui si sa sia chi è il padre sia chi è la madre
- **INTERSECT** restituisce tutti i risultati della prima query che appartengono anche alla seconda query
- Non supportato da MySQL
- Si potrà emulare utilizzando le query annidate

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
```