

SQL: operatori avanzati

Laboratorio di Informatica e Archiviazione

14 Dicembre 2010

Ordinamento

- La clausola `order by`, posta alla fine dell'interrogazione, ordina le tuple del risultato.
- `order by` *Attributo* [`asc` | `desc`]
 , *Attributo* [`asc` | `desc`]
- Estrarre tutti i pazienti, ordinandoli per città e per data di nascita in modo decrescente:

```
select *  
from Paziente  
order by Citta desc, DataNascita desc
```

| ID | Nome | Cognome | DataNascita | Sesso | Indirizzo | Citta |
|----|-----------|-----------|-------------|-------|----------------------------|---------|
| 2 | Paola | Bianchi | 23/5/1985 | F | Corso Cavour, 32 | Torino |
| 3 | Mario | Rossi | 8/11/1943 | M | Piazza Indipendenza, 23 | Torino |
| 5 | Luigi | Di Giusto | 6/7/1984 | M | Corso Vittorio Emanuele, 1 | Roma |
| 4 | Francesca | Anselmi | 8/2/1990 | F | Via Mazzini, 33 | Milano |
| 1 | Giulio | Rossi | 12/3/1965 | M | Via Verdi, 19 | Milano |
| 6 | Marta | Giuliacci | 3/7/1977 | F | Via Garibaldi, 3 | Bologna |

Operatori aggregati

- Sono operatori non rappresentabili nell'algebra relazionale
- Il risultato dipende dai valori di *insiemi di tuple*
- Lo standard SQL prevede cinque operatori aggregati:
 - count
 - sum
 - max
 - min
 - avg

Operatore count

- Restituisce il numero di tuple del risultato, oppure di valori non nulli di un certo attributo

- **Trovare il numero di medici:**

```
select count(*) as NumMedici  
from Medico
```

| NumMedici |
|-----------|
| 7 |

- **Trovare il numero di visite con pressione non nulla:**

```
select count(Pressione) as  
NumVisite  
from Visita
```

| NumVisite |
|-----------|
| 2 |

Somma, media, minimo e massimo

- Sintassi: `sum` | `avg` | `min` | `max` (*espressione*)
- Trovare l'età media dei pazienti:

```
select avg(year(date()) - year(DataNascita))  
        as EtaMedia  
from Paziente
```

| EtaMedia |
|----------|
| 36 |

Join e funzioni aggregate

- Trovare l'età massima dei pazienti che hanno effettuato una visita oculistica:

```
SELECT max(year(date())-year(DataNascita))
        AS EtaMax
FROM Paziente, Visita
WHERE ID=Paziente And Tipo='Oculistica';
```

| EtaMax |
|--------|
| 33 |

Una query sbagliata!

```
SELECT Nome, Cognome,  
       max(year(date())-year(DataNascita))  
       AS EtaMax  
FROM Paziente, Visita  
WHERE ID=Paziente And Tipo='Oculistica';
```

- Di chi è il nome è cognome?
Non si possono mescolare funzioni aggregate e attributi
a meno di non usare l'operatore di raggruppamento

Interrogazioni con raggruppamento

- Permettono di applicare gli operatori aggregati a gruppi di tuple
- “Per ogni città, l’età media dei pazienti”

```
SELECT Città,  
        avg(year(date())-year(DataNascita)) AS EtaMedia  
FROM Paziente  
GROUP BY Città
```

| Città | EtaMedia |
|---------|----------|
| Bologna | 33 |
| Milano | 32,5 |
| Roma | 26 |
| Torino | 46 |

Semantica delle interrogazioni con raggruppamento - 1

- 1 Si esegue l'interrogazione senza `group by` e senza operatori di raggruppamento

```
SELECT Città,  
       year(date())-year(DataNascita) AS EtaMedia  
FROM Paziente
```

| Città | EtaMedia |
|---------|----------|
| Milano | 45 |
| Torino | 25 |
| Torino | 67 |
| Milano | 20 |
| Roma | 26 |
| Bologna | 33 |

Semantica delle interrogazioni con raggruppamento - 2

- 2 Si raggruppano le righe con lo stesso valore per l'attributo di raggruppamento (Città)
- 3 Si applicano le funzioni aggregate in modo separato per ciascun gruppo

| Città | EtaMedia |
|---------|----------|
| Bologna | 33 |
| Milano | 45 |
| Milano | 20 |
| Roma | 26 |
| Torino | 25 |
| Torino | 67 |

| Città | EtaMedia |
|---------|----------|
| Bologna | 33 |
| Milano | 32,5 |
| Roma | 26 |
| Torino | 46 |

Condizioni sui gruppi

- Per imporre condizioni sul risultato di un operatore aggregato si usa la clausola `having`
- “Le città in cui risiedono almeno due pazienti”

```
SELECT Città, Count(*) AS NumPazienti  
FROM Paziente  
GROUP BY Città  
HAVING count(*) >=2
```

| Città | NumPazienti |
|--------|-------------|
| Milano | 2 |
| Torino | 2 |

Having o Where ?

- Solo le condizioni con funzioni aggregate possono stare in `having`
- “I pazienti maschi in cura per almeno due patologie”

```
SELECT Nome, Cognome
FROM Paziente, Cura
WHERE ID=Paziente AND Sesso='M'
GROUP BY Nome, Cognome
HAVING Count (*) >=2
```

| Nome | Cognome |
|--------|---------|
| Giulio | Rossi |
| Mario | Rossi |