

Corso sul linguaggio SQL

Modulo L2B (SQL)

5 - Aggregazione e raggruppamento

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

1
02/01/2015

Prerequisiti

- Conteggio e somma degli elementi di un insieme numerico
- Massimo, minimo e media di un insieme numerico
- Ordinamento degli elementi di un insieme
- Scrivere semplici interrogazioni SQL

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

2
02/01/2015

Introduzione

In questa Unità si vedranno i comandi **SQL** con i quali è possibile raggruppare o ordinare i dati secondo certi criteri e le funzioni che consentono di effettuare conteggi, somme ed altre operazioni.

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

3
02/01/2015

Informazioni generali

N.B. – A solo scopo didattico:

- i caratteri **MAIUSCOLI** indicano parole chiave del linguaggio;
- i caratteri *corsivi* indicano elementi che dovranno essere specificati dal programmatore;
- le parentesi quadre indicano opzione
- la barra verticale “|” indica alternativa.

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

4
02/01/2015

Funzioni di aggregazione

Attraverso il comando **SELECT** è possibile anche di effettuare conteggio, somme ed altre operazioni, come il calcolo del massimo, del minimo e del valore medio. Queste funzioni prendono il nome di **funzioni di aggregazione**.

Abbiamo le seguenti funzioni di aggregazione:

- **COUNT**
- **SUM**
- **AVG**
- **MIN**
- **MAX**

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

5
02/01/2015

Funzione COUNT

La funzione **COUNT** *conteggia* il numero di tuple di una tabella (eventualmente verificanti una condizione). La sintassi è la seguente:

SELECT COUNT(*attrib*)
FROM *tab*;

Conteggia il numero di tuple della tabella *tab* nelle quali è specificato un valore per l'attributo *attrib*.

SELECT COUNT(*)
FROM *tab*;

Conteggia il numero di tuple presenti nella tabella *tab*

SELECT COUNT(*) **AS** *Totale*
FROM *tab*;

Conteggia il numero di tuple della tabella *tab*, visualizzando tale numero in una colonna di nome *Totale*

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

6
02/01/2015

Funzione COUNT

```
SELECT COUNT(*)  
FROM tab  
WHERE cond;
```

Conteggia il numero di tuple della tabella *tab* che verificano la condizione *cond*.

```
SELECT COUNT (DISTINCT attrib)  
FROM tab  
WHERE cond;
```

Conteggia il numero di tuple presenti nella tabella *tab*, aventi valori distinti di *attrib* e che verificano la condizione *cond*.

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

7
02/01/2015

Funzione COUNT

ATTIVITA': Scrivere un comando SQL che sullo schema *Personale* (*Cognome*, *Provincia*, *Livello*) conteggi il numero di persone che hanno "Roma" nell'attributo *Provincia*.

```
SELECT COUNT(*)  
FROM Personale  
WHERE Provincia = "Roma";
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

8
02/01/2015

Funzione COUNT

ATTIVITA': Scrivere un comando SQL che nello schema *Personale* (*Cognome, Provincia, Livello*) conteggi il numero di livelli che hanno "Roma" nell'attributo *Provincia*.

```
SELECT COUNT (DISTINCT Livello)
FROM Personale
WHERE Provincia = "Roma";
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

9
02/01/2015

Funzione SUM

Questa funzione dà la *somma dei valori* di un attributo (eventualmente con condizione). La sintassi è la seguente:

```
SELECT SUM(attrib)
FROM tab;
```

Calcola la somma dei valori contenuti nell'attributo *attrib* della tabella *tab*.

ATTIVITA': Scrivere un comando SQL che nello schema *Film* (*ID, Titolo, Prezzo*) calcoli la somma dei prezzi dei film della tabella *Film*.

```
SELECT SUM(Prezzo)
FROM Film;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

10
02/01/2015

Funzione SUM

ATTIVITA': Scrivere un comando SQL che nello schema *Personale* (*Cognome*, *Provincia*, *Livello*, *StipendioBase*) calcoli la somma degli stipendi base per il solo personale di livello 4.

```
SELECT SUM(StipendioBase)
FROM Personale
WHERE Livello = 4;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

11
02/01/2015

Funzione SUM

ATTIVITA': Scrivere un comando SQL che nello schema *Articolo* (*ID*, *CostoUnitario*, *Quantità*) calcoli la somma totale dei costi di tutti gli articoli, moltiplicando il *CostoUnitario* di ciascuno per la *Quantità* presente e rinominando in *Totale* il valore calcolato.

```
SELECT SUM(CostoUnitario * Quantità) AS Totale
FROM Articolo;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

12
02/01/2015

Funzioni MIN e MAX

Queste funzioni di aggregazione calcolano il valore *minimo* e quello *massimo* di una data colonna. La sintassi per **MIN** (analoga a quella di **MAX**) è:

```
SELECT MIN(attrib)  
FROM tab;
```

Calcola il valore minimo contenuto nella colonna attrib della tabella tab.

ATTIVITA': Scrivere un comando SQL che nella tabella *Film* (*ID*, *Titolo*, *Prezzo*) calcoli il valore minimo della colonna *Prezzo*.

```
SELECT MIN(Prezzo)  
FROM Film;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

13
02/01/2015

Funzione AVG

Questa funzione di aggregazione calcola il *valore medio* tra i valori presenti in una data colonna. La sintassi è:

```
SELECT AVG(attrib)  
FROM tab;
```

Calcola il valore medio contenuto nell'attributo attrib della tabella tab.

ATTIVITA': Scrivere un comando SQL che calcoli il valore medio della colonna *prezzo* della tabella *Film*.

```
SELECT AVG(prezzo)  
FROM Film;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

14
02/01/2015

Raggruppamento

Le funzioni possono essere applicate a **partizioni** (sottoinsiemi di righe) delle relazioni, per calcolarne valori parziali tramite la clausola **GROUP BY**:

La sintassi è la seguente:

```
SELECT .... GROUP BY lista-attributi;
```

Raggruppamento

PATERNITA

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Antonio	Luigi
Marco	Anna

ATTIVITA': Scrivere un comando SQL che per ogni padre nella tabella *Paternita* visualizzi il numero di *figli* posseduti

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

Padre	NumFigli
Antonio	1
Franco	2
Luigi	2
Marco	1
Sergio	1

Ordinamento

Il comando **SELECT** consente anche l'ordinamento dei dati in base al valore di un attributo.

Sintassi:

```
SELECT lista_attributi  
FROM tab  
[WHERE condizione]  
ORDER BY attributo;
```

Ad esempio, l'elenco in ordine alfabetico di *nome* e *reddito* delle persone con meno di trenta anni è dato da:

```
SELECT Nome, Reddito  
FROM Persona  
WHERE Eta < 30 ORDER BY Nome;
```

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

17
02/01/2015

Ordinamento

ATTIVITA': Scrivere un comando SQL che elenchi in ordine alfabetico *Nome* e *Reddito* delle persone con *Reddito* minore o uguale a 30.

```
SELECT Nome, Reddito  
FROM Dipendente  
WHERE Reddito <= 30  
ORDER BY Nome;
```

Nome	Reddito
Aldo	15
Andrea	21
Filippo	30

DIPENDENTE

Nome	Reddito
Filippo	30
Andrea	21
Marco	50
Aldo	15

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

18
02/01/2015

Ordinamento

PATERNITA

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Antonio	Luigi
Marco	Anna

ATTIVITA': Scrivere un comando SQL che elenchi il *Padre* e, in senso crescente, il *numero di figli* posseduti da ciascun padre nella tabella *Paternita*

```
SELECT Padre, COUNT(*) AS NumFigli
FROM Paternita GROUP BY Padre ORDER
BY NumFigli;
```

Padre	NumFigli
Sergio	1
Antonio	1
Marco	1
Luigi	2
Franco	2

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

19
02/01/2015

Ordinamento

PATERNITA

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Antonio	Luigi
Marco	Anna

ATTIVITA': Scrivere un comando SQL che elenchi il *Padre* e, in senso decrescente, il *numero di figli* posseduti da ciascun padre nella tabella *Paternita*

```
SELECT Padre, COUNT(*) AS NumFigli
FROM Paternita GROUP BY Padre ORDER
BY NumFigli DESC;
```

Padre	NumFigli
Luigi	2
Franco	2
Sergio	1
Antonio	1
Marco	1

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

20
02/01/2015

Condizioni sui gruppi

ATTIVITA': Scrivere un comando SQL che, dati gli schemi:

Persona (Nome, Età, Reddito)

Paternita (Padre, Figlio)

elenchi i *Padre* e il *Reddito medio* dei figli aventi un *Reddito medio* maggiore di 25.

```
SELECT Padre, AVG(Reddito)
```

```
FROM Persona JOIN Paternita ON Figlio = Nome
```

```
GROUP BY Padre
```

```
HAVING AVG(Reddito) > 25;
```

La clausola **HAVING** esprime la condizione sui dati aggregati.

Raggruppamenti e target list

scorretta

```
select padre, avg(f.reddito), p.reddito  
from persone f join paternita on figlio = nome join  
    persone p on padre =p.nome  
group by padre
```

corretta

```
select padre, avg(f.reddito), p.reddito  
from persone f join paternita on figlio = nome join  
    persone p on padre =p.nome  
group by padre, p.reddito
```

Condizioni sui gruppi

Attenzione a WHERE e HAVING!

ATTIVITA': Scrivere un comando SQL che, dati gli schemi:

Persona (Nome, Età, Reddito)

Paternita (Padre, Figlio)

elenchi i padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25.

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

Argomenti

- Funzioni di aggregazione
- Funzione **COUNT**
- Funzione **SUM**
- Funzioni **MIN** e **MAX**
- Funzione **AVG**
- Raggruppamento
- Ordinamento
- Condizioni sui gruppi

Altre fonti di informazione

- Atzeni, Ceri, Paraboschi, Torlone, Basi di dati - McGraw-Hill, 1996-2002

M. Malatesta - SQL(5) - Aggregazione e raggruppamenti-15

25
02/01/2015