

[Home Page](#) [Novità](#) [Aiuto](#) 

Il linguaggio SQL Funzioni di aggregazione

http://www.vbsimple.net/database/db_08_07.htm

Difficoltà:  3 / 5

Il termine **funzione di aggregazione** indica quelle funzioni che elaborano un insieme di righe ma restituiscono sempre un risultato; esempi di queste funzioni sono quelle che effettuano la somma dei valori di un campo, il conteggio di righe di un certo tipo, il valore massimo presente nelle righe recuperate. Il loro risultato è sempre uno, indipendentemente dal numero di righe prese in esame. Un'eccezione a questa regola è dettata dall'uso della [clausola GROUP BY](#), che vedremo in seguito per effettuare delle spaccature ed elaborare l'aggregazione sui singoli gruppi.

Sebbene ciascun database possenga le proprie funzioni di aggregazione, ve ne sono alcune così comuni da poter essere considerate ormai uno standard, ed una deficienza per un database che non le supporta. Osserviamo la sintassi generale:

```
<funzione> (<predicato> <campo> )
```

Esse si applicano all'istruzione SELECT e possono apparire nell'elenco dei campi da mostrare (è così nella maggior parte dei casi) oppure nelle clausole [ORDER BY](#) ed [HAVING](#) che vedremo in seguito. Essendo inoltre funzioni e non campi, non possiedono un nome di campo reale, quindi solitamente i risultati vengono rinominati con l'**operatore AS** trattato nelle lezioni precedenti. Prima di vedere le funzioni più comuni in esame presentiamo un esempio generale:

```
SELECT <funzione> (<campo>) AS <nome> FROM <tabella>  
WHERE <criteri>
```

Questa istruzione effettuerebbe il calcolo della funzione, applicata sul campo indicato, per la tabella specificata e per le righe incluse nei criteri; il risultato della funzione sarebbe infine presentato in un campo con il nuovo nome indicato a seguito dell'operatore AS.

La definizione precedente a quest'ultima includeva anche un predicato, posto immediatamente prima del campo per cui effettuare il calcolo. Tale predicato può essere:

- ALL (Valore predefinito se omissivo)
Indica di prendere tutte le righe in esame estratte dalla condizione WHERE
- DISTINCT
Indica di escludere le righe con valori del campo specificato ripetuti e quindi prenderà in esame sempre e soltanto una riga.

Esistono cinque funzioni standard di aggregazione, dall'uso molto semplice e, come già detto, presenti in quasi tutti i database.

La prima di queste funzioni è sicuramente **COUNT**, che effettua il conteggio delle righe estratte e subito vediamo un esempio molto elementare:

```
SELECT COUNT(nome) AS conteggio FROM dipendenti
```

Restituisce il **numero di righe** con valori non **Null** nella colonna nome presenti all'interno dell'intera tabella dipendenti senza applicare altre esclusioni. La funzione opererà sul campo nome e quindi conterà quanti nomi sono presenti all'interno della tabella dipendenti; i valori Null saranno esclusi. Otterremmo lo stesso risultato con la seguente istruzione:

```
SELECT COUNT(ALL nome) AS conteggio FROM dipendenti
```

Ciò in quanto il predicato **ALL** è il predefinito nel caso venisse omissso. Otterremmo invece un risultato differente con questa istruzione:

```
SELECT COUNT(DISTINCT nome) AS conteggio FROM dipendenti
```

Questa restituirà il **numero di nomi differenti** all'interno della tabella dipendenti, escludendo infatti quelle righe con il valore nel campo nome ripetuto. Come nel caso precedente **sono escluse le righe con valore Null** nel campo nome.

Per includere invece tutte le righe, senza escludere né quelle con valore Null né quelle duplicate è possibile specificare il simbolo * al posto del nome del campo ed omettere il predicato **ALL** o **DISTINCT**.

```
SELECT COUNT(*) AS conteggio FROM dipendenti
```

Questa istruzione semplicemente restituisce il numero di righe presenti all'interno della tabella dipendenti. Se avessimo specificato uno dei due predicati, l'istruzione avrebbe riportato un errore. Se si utilizza il simbolo * i predicati non sono necessari; viceversa utilizzando il nome di una colonna può essere utilizzato uno dei due predicati; in assenza di specifica sarà utilizzato il predicato **ALL**.

Prima di procedere oltre vediamo altri due esempi:

```
SELECT COUNT(nome) AS conteggio FROM dipendenti  
WHERE ruolo='Direttore'
```

Restituirà il conteggio dei direttori presenti nell'elenco dei dipendenti; saranno escluse quelle righe in cui il campo nome contiene valore Null, oltre che naturalmente tutte quelle che non corrispondono al ruolo di direttore.

Inoltre le funzioni di aggregazione, come tutti gli altri valori d'altronde, possono essere affiancate da altre funzioni di aggregazione:

```
SELECT COUNT(*) AS totale, COUNT(nome) AS nomi, COUNT(DISTINCT nome) AS  
nomi_unici  
FROM dipendenti WHERE ruolo='Direttore'
```

Restituirà una riga composta da tre campi:

1. **totale** conterrà il numero complessivo di righe della tabella dipendenti in cui il ruolo sia Direttore;
 2. **nomi** conterrà il numero di nomi dei dipendenti, escludendo i valori Null e quelli che non corrispondono a Direttore;
 3. **nomi_unici** conterrà invece il numero di nomi univoci, contando solo una volta quelli duplicati, escludendo quelli (anzi quella, vista l'esclusione dei duplicati) con valori Null e ruolo non corrispondente a Direttore.
-

Un'altra funzione di aggregazione molto comune è **SUM** che, come dice il nome, riporta la somma dei valori presi in esame; come per tutte le altre funzioni di aggregazione il procedimento è sempre lo stesso, utilizzando il nome di un campo e un predicato ALL o DISTINCT. A differenza della precedente funzione, questa e le successive non **consentono di specificare l'asterisco** per indicare tutti i valori del criterio e pertanto i valori Null sono sempre esclusi. Vediamo qualche esempio di funzionamento:

```
SELECT SUM(stipendio) AS somma FROM dipendenti
WHERE ruolo='Impiegati'
```

Riporterà la somma dei dati presenti nella colonna stipendio della tabella dipendenti in cui il ruolo corrisponde a Impiegati. Sono esclusi tutti i valori Null.

Questa funzione lavorerà soltanto su valori numerici interi o decimali e restituirà errore in caso di dati di tipo carattere.

Molto simile alla precedente è la funzione **AVG**, utilizzata per calcolare il valore medio di un campo; opera soltanto su valori numerici ed il suo funzionamento è molto semplice:

```
SELECT AVG(stipendio) AS media FROM dipendenti
WHERE ruolo='Impiegati'
```

Restituirà lo stipendio medio di tutti gli impiegati presenti nella tabella dipendenti. Non avendo specificato il predicato sarà assunto il valore predefinito ALL che include tutte le righe nel calcolo. Specificando invece il predicato DISTINCT otterremo un calcolo differente:

```
SELECT AVG(DISTINCT stipendio) AS media FROM dipendenti
WHERE ruolo='Impiegati'
```

Sarà infatti riportato il valore dello stipendio medio per mansione degli impiegati; cercherò di essere più chiaro: se la tabella dipendenti contenesse 10 impiegati pagati 1000 e 3 impiegati pagati 3000 il valore restituito sarà calcolato come media tra 1000 e 3000, riportando quindi il valore di 2000. Se non avessimo specificato il predicato DISTINCT il calcolo avrebbe restituito il valore $1000 \cdot 10 + 3000 \cdot 3$ risultando quindi il valore di 1461.

È importante capire questa differenza perché potrebbe produrre risultati inaspettati.

Restano le ultime due funzioni di aggregazione: **MIN** e **MAX** che, come dicono i loro nomi, restituiscono rispettivamente il valore minimo ed il valore massimo dell'insieme richiesto. Queste due funzioni lavorano bene sia su campi numerici che su campi di tipo carattere.

```
SELECT MIN(nome) AS primo, MAX(nome) AS ultimo FROM dipendenti
```

Restituirà il primo e l'ultimo alfabeticamente dei dipendenti, cioè quello in cui il campo nome apparirà in prima posizione e quello che sarà in ultima se le righe venissero ordinate.

Come per le altre funzioni è possibile specificare il predicato **ALL** o **DISTINCT** ma in entrambi i casi di queste due funzioni il loro effetto è inutile; l'inclusione o l'esclusione non comporta alcuna differenza; saranno sempre e comunque recuperati il primo e l'ultimo.

```
SELECT MIN(stipendio) AS povero, MAX(stipendio) AS ricco FROM dipendenti
```

Naturalmente è possibile effettuare la stessa interrogazione anche su dati numerici e, nel nostro esempio, recuperare lo stipendio minimo e lo stipendio massimo di tutti i dipendenti.

[Eibia FBI](#)
15 Marzo 2004



[Torna all'indice della sezione Database](#)
