

SQL Data Manipulation Language

Raggruppamenti

DML – Raggruppamenti

Le istruzioni di SELECT viste in precedenza permettono di recuperare informazioni che si riferiscono a singole righe.

Ci sono molti casi nei quali è necessario recuperare informazioni riepilogative di gruppi di righe.

Per esempio:

- Il numero di atleti italiani che hanno partecipato alle olimpiadi di Londra del 2012.
- Il numero di olimpiadi disputate in ogni città

Per soddisfare queste necessità SQL mette a disposizione due strumenti:

- Funzioni di aggregazione
- Clausola GROUP BY

DML – Funzioni di Aggregazione

Le funzioni di aggregazione sono funzioni, principalmente matematiche, che lavorano su una intera colonna (o su un gruppo, lo vedremo tra poco) e restituiscono un unico valore anziché tutte le righe previste dalla clausola WHERE.

```
SELECT SUM(Stipendio) As StipendioMensileTotale  
FROM Dipendenti  
WHERE Ruolo = 'Medico'
```

La query di esempio genera una sola riga di risultato, con un solo campo, contenente la somma del campo Stipendio calcolata su tutte le righe che soddisfano la clausola WHERE.

DML – Funzioni di Aggregazione

Le funzioni di aggregazione possono anche lavorare su espressioni applicate a campi

```
SELECT SUM(Stipendio*12) As StipendioTotale  
FROM Dipendenti  
WHERE Ruolo = 'Medico'
```

DML – Funzioni di Aggregazione

Le principali funzioni di aggregazione sono:

- MIN → Minimo
- MAX → Massimo
- SUM → Somma
- AVG → Media aritmetica
- COUNT → Conteggio di righe

Solamente COUNT considera i valori NULL, tutte le altre funzioni li ignorano.

Min e Max funzionano anche su campi testuali e considerano l'ordinamento alfabetico.

Sum e Avg funzionano solo su campi numerici, mentre Count funziona su qualsiasi tipo di dato.

DML – Funzioni di Aggregazione

All'interno della stessa query posso inserire più funzioni di aggregazione. Verranno calcolate in modo indipendente l'una dall'altra

```
SELECT SUM(Stipendio),MIN(Stipendio),MAX(Stipendio),AVG(Stipendio),  
COUNT(Stipendio)  
FROM Dipendenti  
WHERE Ruolo = 'Medico'
```

La query calcola la somma degli stipendi, lo stipendio minimo, massimo e medio dei dipendenti con ruolo Medico. Calcola inoltre il numero di righe su cui i risultati sono calcolati.

Non possono essere inseriti campi che non siano inseriti in funzioni di aggregazione

DML – Funzioni di Aggregazione - COUNT

La funzione COUNT è l'unica che considera i valori NULL. E' possibile comunque specificare se considerarli oppure no:

- COUNT(*) → I valori NULL vengono considerati
- COUNT(NomeCampo) → I valori NULL non vengono considerati

`SELECT COUNT(*) FROM Athletes` → Numero totale di record nella tabella

`SELECT COUNT(Medal) FROM Athletes` → Numero di record per i quali Medal non è null. E' pertanto equivalente a:

`SELECT COUNT(*) FROM Athletes WHERE Medal IS NOT NULL`

DML – Funzioni di Aggregazione - DISTINCT

E' possibile fare in modo che le funzioni di aggregazione lavorino su valori distinti. E' sufficiente inserire la parola chiave DISTINCT all'interno della funzione:

```
SELECT COUNT(DISTINCT Sport) FROM Athletes
```

La query restituisce il numero totale di sport mai apparsi alle olimpiadi

DML – CAST

Le funzioni di aggregazione determinano in automatico il tipo del risultato:

- SUM su colonna INT → Risultati di tipo INT
- SUM su colonna FLOAT → Risultato FLOAT
- AVG su colonna INT → Risultato INT

Per impostare in modo arbitrario il tipo di dato si utilizza la funzione CAST, specificando il tipo desiderato.

```
SELECT CAST(COUNT(DISTINCT Sport) AS FLOAT) FROM Athletes
```

La funzione CAST è utilizzabile anche nella clausola SELECT per modificare il tipo di dato delle colonne calcolate.

Esercizio: Funzioni di aggregazione

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

1. Quante città hanno ospitato giochi olimpici?
2. Quanti atleti hanno vinto la medaglia d'oro?
3. Qual è l'età media dei partecipanti?
4. Qual è l'età media dei medagliati?
5. Qual è l'età dell'atleta medagliato più giovane? (farla con aggregazione)
6. Quanti atleti italiani hanno Rossi nel cognome? (anche De Rossi, ma non Rossini ne Grossi)
7. Qual è l'altezza media degli atleti di basket maschile? E della ginnastica femminile?

Esercizio: Funzioni di aggregazione - Soluzioni

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

1. Quante città hanno ospitato giochi olimpici?

```
SELECT COUNT(DISTINCT City) FROM Athletes
```

2. Quanti atleti hanno vinto la medaglia d'oro?

```
SELECT COUNT(DISTINCT IdAthlete) FROM Athletes WHERE Medal = 'Gold'
```

3. Qual è l'età media dei partecipanti?

```
SELECT AVG(CAST(Age AS FLOAT)) FROM Athletes
```

4. Qual è l'età media dei medagliati?

```
SELECT AVG(CAST(Age AS FLOAT)) FROM Athletes WHERE Medal IS NOT NULL
```

Esercizio: Funzioni di aggregazione

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

5. Qual è l'età dell'atleta medagliato più giovane? (farla con aggregazione)

```
SELECT MIN(Age) FROM Athletes WHERE Medal IS NOT NULL
```

6. Quanti atleti italiani hanno Rossi nel cognome? (anche De Rossi, ma non Rossini ne Grossi)

```
SELECT COUNT(DISTINCT IdAthlete) FROM Athletes  
WHERE NOC = 'ITA' AND (Name LIKE '% Rossi' OR Name LIKE '% Rossi %' OR Name LIKE 'Rossi %')
```

7. Qual è l'altezza media degli atleti di basket maschile? E della ginnastica femminile?

```
SELECT AVG(Height) FROM Athletes WHERE Sex='M' AND Sport = 'Basketball'  
SELECT AVG(Height) FROM Athletes WHERE Sex='F' AND Sport = 'Gymnastics'
```

DML – Raggruppamento

Le funzioni di aggregazione viste fino ad ora non sono ancora sufficienti per coprire tutti i casi. Infatti:

- Calcolano il risultato su tutte le righe che soddisfano la clausola WHERE
- Non permettono di inserire campi non aggregati

Per superare questi limiti si utilizza la clausola GROUP BY, seguita da un elenco di campi su cui raggruppare. Per esempio:

```
SELECT NOC,COUNT(*) FROM Athletes GROUP BY NOC
```

In questa query viene restituito, per ogni nazione, il numero di atleti che hanno preso parte alle olimpiadi. Le righe vengono raggruppate a parità di campo NOC. Per ogni gruppo di righe viene calcolata la funzione di aggregazione COUNT.

I campi inseriti nella clausola GROUP BY possono anche essere inseriti in SELECT. Essendo identici all'interno del singolo gruppo non possono dare origine ad ambiguità.

DML – Raggruppamento

Una query con group by ritorna tante righe quanti sono i gruppi generati. E' possibile effettuare un ulteriore filtro per restituire solamente le righe che soddisfano determinate condizioni.

Per ottenere questo risultato si utilizza la clausola HAVING, nella quale si possono utilizzare le stesse modalità previste per WHERE potendo però inserire solamente campi aggregati e funzioni di aggregazione.

```
SELECT NOC,COUNT(Medal) FROM Athletes  
WHERE Year = 2016  
GROUP BY NOC  
HAVING COUNT(Medal) >5
```

La query restituisce le nazioni che, nel 2016, hanno vinto più di 5 medaglie, riportandone anche il numero totale.

DML – Raggruppamento – Having e Where

Le clausole HAVING e WHERE hanno una funzionalità simile ma restano comunque differenti:

- WHERE → Filtra le righe prima del raggruppamento
- HAVING → Filtra le righe raggruppate dopo il raggruppamento

Il filtro per campo di raggruppamento può essere fatto sia su WHERE che su HAVING. Le due sintassi che seguono sono equivalenti:

```
SELECT NOC, COUNT (*)  
FROM Athletes  
GROUP BY NOC  
HAVING NOC = 'ITA'
```

```
SELECT NOC, COUNT (*)  
FROM Athletes  
WHERE NOC = 'ITA'  
GROUP BY NOC
```

DML – Raggruppamento e ordinamento

L'ordinamento avviene con la solita clausola ORDER BY ma avviene dopo il raggruppamento e pertanto la clausola va scritta dopo GROUP BY e HAVING.

Nella clausola ORDER BY è possibile però inserire solamente campi aggregati e funzioni di aggregazione.

```
SELECT NOC,COUNT(Medal) FROM Athletes
```

```
WHERE Year = 2016
```

```
GROUP BY NOC
```

```
HAVING COUNT(Medal)>5
```

```
ORDER BY NOC ASC, COUNT(*) DESC
```

```
(ORDER BY 1 ASC, 2 DESC)
```


Esercizio: Funzioni di aggregazione con raggruppamento

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

1. Per ogni città calcolare quante edizioni ha ospitato. Ordinare in ordine decrescente.
2. Recuperare i 10 atleti che hanno vinto più medaglie d'oro, calcolandone il numero ed ordinando in modo decrescente.
3. Qual è l'età media dei medagliati per ogni edizione disputata. Ordinare per media crescente.
4. Ordinare le edizioni per numero totale, decrescente, di medaglie vinte dall'Italia.
5. Quale edizione ha visto la partecipazione di più nazioni?
6. Stilare la classifica degli sport che hanno assegnato più medaglie.
7. Stilare l'elenco degli atleti che hanno vinto più di 3 medaglie in almeno due edizioni
8. Stilare l'elenco degli atleti con il maggior tempo tra prima e ultima medaglia vinte.

Esercizio: Funzioni di aggregazione con raggruppamento - Soluzioni

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

1. Per ogni città calcolare quante edizioni ha ospitato. Ordinare in ordine decrescente.

```
SELECT City, COUNT(DISTINCT Games) AS Editions  
FROM Athletes GROUP BY City  
ORDER BY COUNT(DISTINCT Games) DESC, City
```

2. Recuperare i 10 atleti che hanno vinto più medaglie d'oro, calcolandone il numero ed ordinando in modo decrescente.

```
SELECT TOP(10) IdAthlete, Name, COUNT(Medal) AS Golds  
FROM Athletes WHERE Medal = 'Gold'  
GROUP BY IdAthlete, Name  
ORDER BY COUNT(Medal) DESC, Name
```

Esercizio: Funzioni di aggregazione con raggruppamento - Soluzioni

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

3. Qual è l'età media dei medagliati per ogni edizione disputata. Ordinare per media crescente.

```
SELECT Games, AVG(Age) AS AvgAge
FROM Athletes WHERE Medal IS NOT NULL
GROUP BY Games
ORDER BY AVG(Age)
```

4. Ordinare le edizioni per numero totale, decrescente, di medaglie vinte dall'Italia.

```
SELECT Games, COUNT(Medal) AS Medals
FROM Athletes WHERE NOC = 'ITA'
GROUP BY Games
ORDER BY COUNT(Medal) DESC
```

Esercizio: Funzioni di aggregazione con raggruppamento - Soluzioni

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

5. Quale edizione ha visto la partecipazione di più nazioni?

```
SELECT TOP(1) Games, COUNT(DISTINCT NOC) AS Nations  
  
FROM Athletes  
  
GROUP BY Games  
  
ORDER BY COUNT(DISTINCT NOC) DESC
```

6. Stilare la classifica degli sport che hanno assegnato più medaglie.

```
SELECT Sport, COUNT(Medal) AS Medals  
  
FROM Athletes  
  
GROUP BY Sport  
  
ORDER BY COUNT(Medal) DESC
```

Esercizio: Funzioni di aggregazione con raggruppamento - Soluzioni

Con riferimento alla tabella Athletes del db Olympics, scrivere query per dare risposta alle seguenti domande:

7. Stilare l'elenco degli atleti che hanno vinto più di 3 medaglie in almeno due edizioni

```
SELECT IdAthlete, Name, COUNT(Medal) AS Medals, COUNT(DISTINCT Games) AS Partecipations
FROM Athletes
WHERE Medal IS NOT NULL
GROUP BY IdAthlete, Name
HAVING COUNT(Medal)>3 AND COUNT(DISTINCT Games)>=2
ORDER BY COUNT(Medal) DESC
```

8. Stilare l'elenco degli atleti con il maggior tempo tra prima e ultima medaglia vinte.

```
SELECT IdAthlete, Name, Max(Year) - Min(Year) AS TimeDifference
FROM Athletes WHERE Medal IS NOT NULL
GROUP BY IdAthlete, Name
ORDER BY Max(Year) - Min(Year) DESC
```