



Linguaggio SQL

terza parte

Funzioni di aggregazione (1)

- **COUNT, SUM, MIN, MAX, AVG, ...**
- Sintetizzano le informazioni di una colonna in un solo valore
- Possono comparire solo nelle clausole **SELECT** (e **HAVING** che vedremo in seguito)
- **COUNT**: conta il numero di righe o di valori non nulli in una colonna
 - **COUNT (*)** conta le righe di una tabella
- **SUM**: somma i valori non nulli di una colonna
- **AVG**: restituisce la media dei valori non nulli di una colonna
 - **AVG (Stipendio) = SUM (Stipendio) / COUNT (Stipendio)**
- **MIN, MAX**: restituiscono il valore minimo e massimo di una colonna

```
SELECT COUNT(*) FROM Impiegati;
```

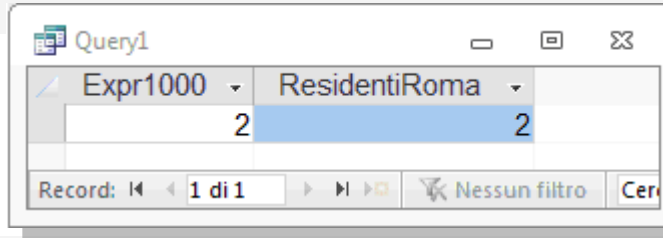
Restituisce 12

Funzioni di aggregazione (2)

```
SELECT COUNT(Dipartimento) FROM Impiegati;
```

Restituisce 11

```
SELECT COUNT(*) AS ResidentiRoma  
FROM Impiegati  
WHERE Residenza = 'Roma';
```

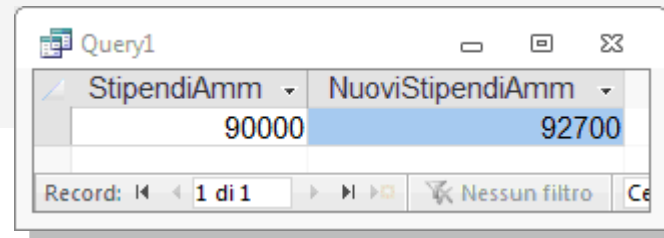


Expr1000	ResidentiRoma
2	2



Field	Type
ID	int(2)
Nome	varchar(10)
Cognome	varchar(7)
Residenza	varchar(8)
Stipendio	decimal(9,2)
Dipartimento	varchar(5)

```
SELECT SUM(Stipendio) AS StipendiAmm,  
       SUM(Stipendio*1.03) AS NuoviStipendiAmm  
FROM Impiegati  
WHERE Dipartimento = 'Amm';
```



StipendiAmm	NuoviStipendiAmm
90000	92700

Funzioni di aggregazione (3)

```
SELECT AVG(Stipendio)
FROM Impiegati
WHERE Residenza = 'Torino';
```

AVG(Stipendio)
32666.666667

```
SELECT ROUND(AVG(Stipendio))
FROM Impiegati
WHERE Residenza = 'Torino';
```

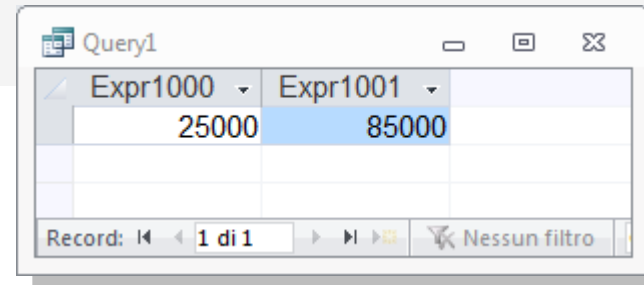
ROUND(AVG(Stipendio))
32667

- Per arrotondare i risultati si usa la funzione **ROUND**

ROUND(*Espressione*, *NumCifre*)

Funzioni di aggregazione (4)

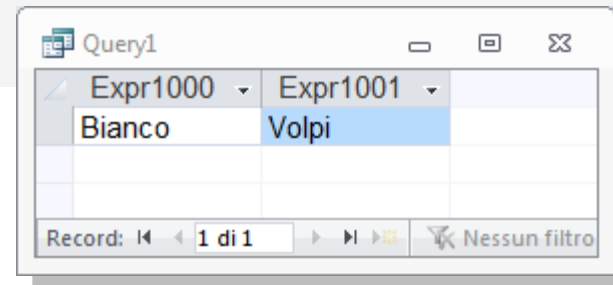
```
SELECT MIN(Stipendio), MAX(Stipendio)  
FROM Impiegati;
```



Expr1000	Expr1001
25000	85000

Record: 1 di 1 Nessun filtro

```
SELECT MIN(Cognome), MAX(Cognome)  
FROM Impiegati;
```



Expr1000	Expr1001
Bianco	Volpi

Record: 1 di 1 Nessun filtro

Ordinamenti e Raggruppamenti

Ordinamenti (1)

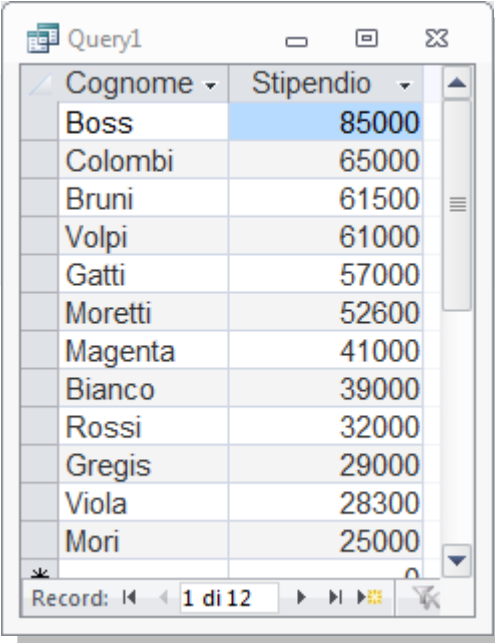
ORDER BY, se presente, deve essere l'ultima clausola di **SELECT**

```
SELECT Cognome, Nome, Residenza  
FROM Impiegati  
ORDER BY Cognome, Nome;
```

Ordinato per valori crescenti
di Cognome e, a parità di
Cognome, per Nome

```
SELECT Cognome, Stipendio  
FROM Impiegati  
ORDER BY Stipendio DESC, Cognome;
```

ORDER BY *Colonna* ASC | DESC



Cognome	Stipendio
Boss	85000
Colombi	65000
Bruni	61500
Volpi	61000
Gatti	57000
Moretti	52600
Magenta	41000
Bianco	39000
Rossi	32000
Gregis	29000
Viola	28300
Mori	25000

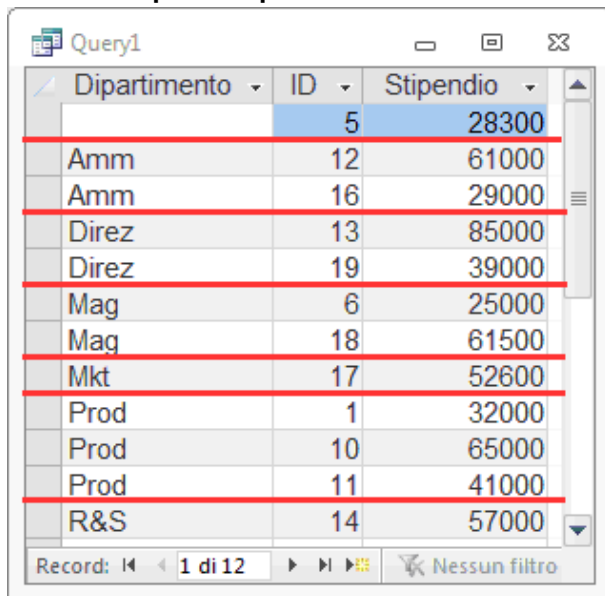
Raggruppamenti (1)

- GROUP BY** per sintetizzare i valori di un campo per classi omogenee

```
SELECT Dipartimento, COUNT(ID) AS .. , SUM(Stipendio) AS ..  
FROM Impiegati  
GROUP BY Dipartimento;
```

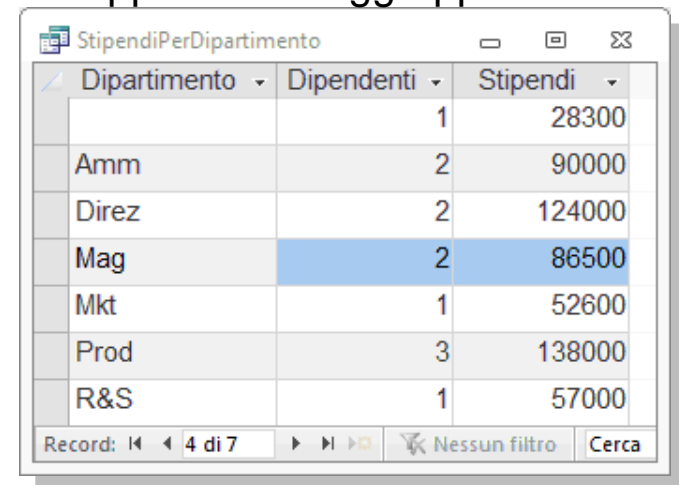
Nella clausola **SELECT**
possono comparire solo i campi
elencati in **GROUP BY** e
funzioni di aggregazione

Le righe sono raggruppate
per dipartimento



Dipartimento	ID	Stipendio
	5	28300
Amm	12	61000
Amm	16	29000
Direz	13	85000
Direz	19	39000
Mag	6	25000
Mag	18	61500
Mkt	17	52600
Prod	1	32000
Prod	10	65000
Prod	11	41000
R&S	14	57000

Le funzioni di aggregazione sono
applicate ai raggruppamenti



Dipartimento	Dipendenti	Stipendi
	1	28300
Amm	2	90000
Direz	2	124000
Mag	2	86500
Mkt	1	52600
Prod	3	138000
R&S	1	57000



Condizioni sui raggruppamenti (1)

- HAVING** per elencare i dipartimenti con più di due dipendenti

```
SELECT Dipartimento, COUNT(*), SUM(Stipendio)
FROM Impiegati
GROUP BY Dipartimento
HAVING COUNT(*) > 2;
```

The screenshot shows the Microsoft Access interface. On the left, the 'Impiegati' table is shown with fields: ID (primary key), Nome, Cognome, Residenza, Stipendio, and Dipartimento. On the right, a summary view of 'Query2' is displayed. The summary view has columns: Dipartimento, ConteggioDiC (Count of ID), and SommaDiStipendio (Sum of Stipendio). The data row shows 'Prod' with a count of 3 and a sum of 138000. Below the summary view, a table with 5 columns is shown: Campo, Tabella, Formula, Ordinamento, and Mostra. The 'Formula' column contains the expression 'COUNT(*) > 2', which is highlighted with a red box. A yellow callout box with the text 'HAVING predica sui raggruppamenti' points to this formula. A red dashed line also connects the 'HAVING COUNT(*) > 2;' part of the SQL query to the formula in the table.

Campo	Tabella	Formula	Ordinamento	Mostra
Dipartimento	Impiegati	Raggruppamento		<input checked="" type="checkbox"/>
ID	Impiegati	Conteggio		<input checked="" type="checkbox"/>
Stipendio	Impiegati	Somma		<input checked="" type="checkbox"/>
[ID]	Impiegati	Conteggio		<input type="checkbox"/>

Condizioni sui raggruppamenti (2)

- **HAVING** e **WHERE**: i dipartimenti di Torino con più di 1 dipendente

```
SELECT Residenza, count(*)  
FROM impiegati  
WHERE Stipendio > 30000  
GROUP BY Residenza  
HAVING count(*) > 1;
```

WHERE predica
sulle righe

HAVING predica sui
raggruppamenti

Firenze	2
Torino	2

Il comando SELECT

- Tutte le clausole del comando **SELECT**

SELECT *Elenco di espressioni da mostrare*
FROM *Tabelle da cui estrarre le righe*
WHERE *Condizioni sulle congiunzioni e sulle righe estratte*
GROUP BY *Campi da considerare per i raggruppamenti*
HAVING *Condizioni sui raggruppamenti*
ORDER BY *Ordinamenti sulle espressioni elencate in SELECT*

- Obbligatorie le sole clausole **SELECT** (e **FROM** se ci sono tabelle)
- Devono rispettare l'ordinamento: **SELECT → FROM → ... → ORDER BY**
- **SELECT** è valutata dal motore del DBMS nell'ordine:
FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY