

Informazioni di sintesi

> Quanto sinora visto permette di estrarre dal DB informazioni che si riferiscono a singole tuple (eventualmente ottenute mediante operazioni di join).

Esempio: il ruolo dell'impiegato 'E001', il responsabile della sede '502', il ruolo degli impiegati della sede '501', ecc.

➤ In molti casi è viceversa utile ottenere dal DB informazioni (di sintesi) che caratterizzano "gruppi" di tuple.

Esempio: il numero di programmatori della sede 'SO1', la media degli stipendi a Bologna, ecc.

- > A tale scopo SQL mette a disposizione due strumenti di base:
 - funzioni aggregate:
 - clausola GROUP BY.





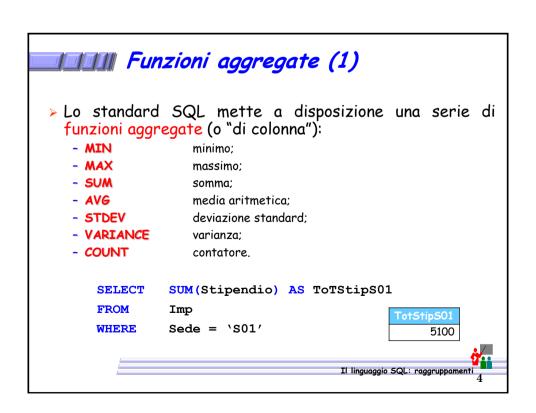
PO1

P02

Bologna

Bologna

Il linguaggio SQL: raggruppamenti



Funzioni aggregate (2)

 L'argomento di una funzione aggregata è una qualunque espressione che può figurare nella SELECT list (ma NON un altra funzione aggregata!)

SELECT SUM(Stipendio*12) AS ToTStipAnnuiS01

FROM Imp

WHERE Sede = 'S01'

TotStipAnnuiS01 61200

- > Tutte le funzioni, ad eccezione di COUNT, ignorano i valori nulli.
- > Il risultato è NULL se tutti i valori sono NULL.
- L'opzione DISTINCT considera solo i valori distinti:

SELECT SUM (DISTINCT Stipendio)

FROM Imp

WHERE Sede = 'S01'

4100

Il linguaggio SQL: raggruppamenti

COUNT e valori nulli

La forma COUNT(*) conta le tuple del risultato; viceversa, specificando una colonna, si omettono quelle con valore nullo in tale colonna:

Imp

•				
CodImp	Sede		Stipendio	
E001	501		2000	
E002	502		1500	
E003	501		1000	
E004	503		NULL	
E005	502		2500	
E006	501		NULL	
E007	501		1000	
E008	502		1200	

SELECT COUNT (*) AS NumImpS01

FROM Imp

WHERE Sede = 'S01'

NumImpS01

SELECT COUNT (Stipendio)

AS NumStipS01

FROM Imp

WHERE Sede = 'S01'

NumStipS01

Il linguaggio SQL: raggruppamenti

Funzioni aggregate e tipo del risultato

> Per alcune funzioni aggregate, al fine di ottenere il risultato desiderato, è necessario operare un *casting* dell'argomento:

SI)	Imp
FI	Stipendio	
	2000	
	1500	
	1000	
	1000	
SI	2500	
	1100	
	1000	
	1200	
•		

SELECT	AVG (Stipe	endio) AS	AvgSti	.p
FROM	Imp	valore	esatto	1412.5

AvgStip 1412

SELECT AVG(CAST(Stipendio AS Decimal(6,2)))
AS AvgStip FROM Imp

AvgStip 1412.50

Il linguaggio SQL: raggruppamenti

Clausola SELECT e funzioni aggregate

> Se si usano funzioni aggregate, la SELECT list <u>non può includere altri</u> <u>elementi</u> che non siano a loro volta funzioni aggregate:

SELECT Nome, MIN(Stipendio)
FROM Imp

non va bene!

(viceversa, SELECT MIN(Stipendio), MAX(Stipendio).. è corretto)

- Il motivo è che una funzione aggregata restituisce un singolo valore, mentre il riferimento a una colonna è in generale un insieme di valori (eventualmente ripetuti).
- > Nel caso specifico (chi sono gli impiegati con stipendio minimo?) è necessario ricorrere a un'altra soluzione, che vedremo più avanti.



🚛 Funzioni aggregate e raggruppamento

- I valori di sintesi calcolati dalle funzioni aggregate si riferiscono a tutte le tuple che soddisfano le condizioni delle clausola WHERE.
- Viceversa in molti casi è opportuno fornire i suddetti valori per gruppi omogenei di tuple (es: impiegati di una stessa sede).
- La clausola GROUP BY serve a definire tali gruppi, specificando una o più colonne (di raggruppamento) sulla base della/e quale/i le tuple sono raggruppate per valori uguali.

SELECT Sede, COUNT(*) AS NumProg FROM amI

WHERE Ruolo = 'Programmatore' **GROUP BY Sede**

Sede	NumProg
501	2
503	1
502	1

> La SELECT list può includere solo le colonne di raggruppamento, ma non altre!

Il linguaggio SQL: raggruppamenti

I■■■Come si ragiona con il GROUP BY

> Le tuple che soddisfano la clausola WHERE...

CodImp	Nome	Sede	Ruolo	Stipendio
E003	Bianchi	501	Programmatore	1000
E004	Gialli	503	Programmatore	1000
E007	Violetti	501	Programmatore	1000
E008	Aranci	502	Programmatore	1200

...sono raggruppate per valori uguali della/e colonna/e presenti nella clausola GROUP BY...

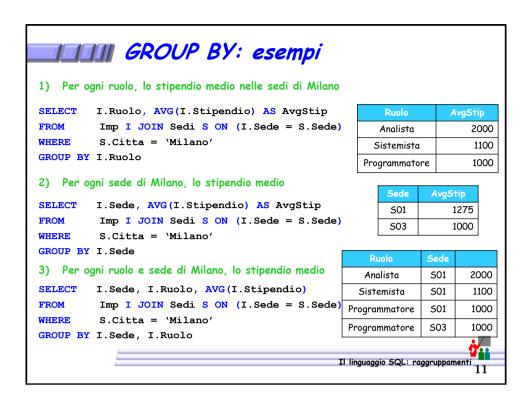
CodImp	Nome	Sede	Ruolo	Stipendio
E003	Bianchi	501	Programmatore	1000
E007	Violetti	501	Programmatore	1000
E004	Gialli	503	Programmatore	1000
E008	Aranci	502	Programmatore	1200

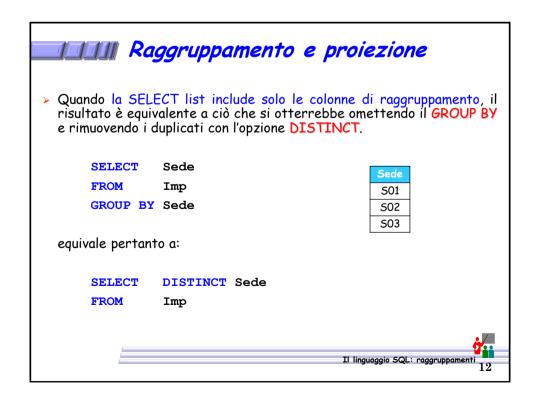
> ...e infine a ciascun gruppo si applica la funzione aggregata.

Sede	NumProg
501	2
503	1
502	1



Il linguaggio SQL: raggruppamenti





🍱 Condizioni sui gruppi

Oltre a poter formare dei gruppi, è anche possibile selezionare alcuni gruppi sulla base di loro proprietà "complessive":

```
        SELECT
        Sede, COUNT(*) AS NumImp

        FROM
        Imp
        Sede NumImp

        GROUP BY Sede
        501 4

        HAVING
        COUNT(*) > 2
        502 3
```

La clausola HAVING ha per i gruppi una funzione simile a quella che la clausola WHERE ha per le tuple (attenzione a non confonderle!).



IIIII Tipi di condizioni sui gruppi

- Nella clausola HAVING si possono avere due tipi di predicati:
 - predicati che fanno uso di funzioni aggregate (es. COUNT (*) > 2);
 - predicati che si riferiscono alle colonne di raggruppamento; questi ultimi si possono anche inserire nella clausola WHERE.

```
Sede, COUNT(*) AS NumImp
SELECT
FROM
GROUP BY Sede
HAVING
         Sede <> 'S01'
                                              Sede
                                                   NumImp
                                              502
                                                         3
                                              503
                                                         1
equivale a:
SELECT
          Sede, COUNT(*) AS NumImp
FROM
```

Sede <> 'S01'

WHERE

GROUP BY Sede

Un esempio completo

Per ogni sede di Bologna in cui il numero di impiegati è almeno 3, si vuole conoscere il valor medio degli stipendi, ordinando il risultato per valori decrescenti di stipendio medio e quindi per sede

```
I.Sede, AVG(Stipendio) AS AvgStipendio
SELECT
FROM
         Imp I, Sedi S
WHERE
         I.Sede = S.Sede
         S.Citta = 'Bologna'
GROUP BY I.Sede
         COUNT(*) >= 3
HAVING
ORDER BY AvgStipendio DESC, Sede
```



L'ordine delle clausole è sempre come nell'esempio.

Si ricorda che il GROUP BY non implica alcun ordinamento del risultato.

Il linguaggio SQL: raggruppamenti

1111 Sommario:

- > Le funzioni aggregate di SQL permettono di ottenere informazioni di sintesi sulle tuple che soddisfano la clausola WHERE.
- Mediante la clausola GROUP BY è possibile suddividere tali tuple in gruppi, per ognuno dei quali si possono quindi calcolare informazioni di sintesi.
- Se le informazioni non sono richieste per tutti i gruppi, si ricorre alla clausola HAVING, che permette di esprimere condizioni a livello di gruppo (anziché di singola tupla).

