Linguaggio SQL

- DML: funzioni aggregate
- DML: interrogazioni con raggruppamento

Database di esempio

C
•
J

<u>SNum</u>	SName	Status	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S 4	Clark	20	London
S5	Adams	30	Athens

SP

<u>SNum</u>	PNum PNum	QTY
S1	P1	300
S1	P2	200
S1	Р3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S 4	P2	200
S 4	P4	300
S 4	P5	400

P

<u>PNum</u>	PName	Color	Weight	City
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
Р3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

- Finora abbiamo considerato esclusivamente condizioni in cui i predicati vengono valutati sulle singole tuple, indipendentemente le une dalle altre
- In SQL è possibile valutare proprietà e condizioni su gruppi di righe
- Le funzioni aggregate prendono in considerazione gruppi di righe e restituiscono un (unico) valore per ogni gruppo di righe
- Le funzioni aggregate più usate sono quelle di conteggio, somma, massimo, minimo

- Esempio: estrarre il numero di parti rosse
- Posso usare la funzione aggregata count in questo modo:

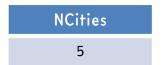
```
select count(*)
from P
where Color = 'Red';
```

- count(*) conta il numero di righe (compresi eventuali valori nulli e duplicati)
- · Posso ovviamente anche rinominare la colonna

```
select count(*) RedPartCount
from P
where Color = 'Red';
RedPartCount
```

• Esempio: contare il numero di città in cui vi sono fornitori (compresi duplicati)

select count(City) NCities
from S;



- select count(Attr) conta il numero di valori non nulli dell'attributo Attr compresi i duplicati
- Per contare valori <u>distinti</u> posso usare <u>distinct</u>
 <u>select count(distinct City) NCities</u>

from S;

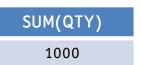
- · Posso anche contare i valori su più attributi
- Esempio: estrarre il numero di nomi (distinti) dei fornitori e il numero di città (distinte)

NNames	NCities
5	3

- Funzioni disponibili:
 - count(): conteggio degli elementi della colonna attr
 - count(*) conteggio delle righe di una tabella
 - count(attr) conteggio dei valori non nulli di attr
 - sum(attr): somma dei valori nella colonna attr
 - avg(attr): media dei valori della colonna attr
 - sum e avg si applicano solo a valori numerici
 - max(attr): massimo valore nella colonna attr
 - min(attr): minimo valore nella colonna attr
 - I DBMS forniscono di solito diverse altre funzioni (es. deviazione standard)
- L'argomento della funzione può essere preceduto da distinct (PostgreSQL non supporta count(distinct *))

• Esempio 1: estrarre la quantità totale di parti *P2* fornite

```
select sum(QTY)
from SP
where PNum = 'P2';
```



• Esempio 2: estrarre il primo fornitore in ordine alfabetico

```
select min(SName)
from S;
```



- È possibile utilizzare più funzioni anche sullo stesso attributo
 - Esempio: estrarre la quantità media, minima e massima di tutte le forniture

```
select avg(QTY), min(QTY), max(QTY) from SP;
```

AVG(QTY)	MIN(QTY)	MAX(QTY)
258,333333333	100	400

- Una funzione aggregata viene valutata su un gruppo di righe e restituisce un unico valore per ogni gruppo
- Non è possibile usare funzioni aggregate (che considerano gruppi di righe) direttamente nella clausola where (che viene valutata per ogni riga)
- select SName from S where Status = max(Status);

- Una funzione aggregata viene valutata su un gruppo di righe e restituisce un unico valore per ogni gruppo
- Non è possibile combinare funzioni aggregate (che restituiscono un unico valore) con attributi non aggregati che possono assumere valori diversi nello stesso gruppo
 - select SName, max(Status) from S;

- SQL non gestisce la possibile differenza di cardinalità tra le funzioni aggregate e le altre espressioni
- Per evitare possibili errori, query di questo tipo vengono considerate dai DBMS scorrette a priori

- È possibile suddividere le righe di una tabella in più gruppi e applicare la funzione aggregata a ogni gruppo
- Ad esempio: non mi interessa il numero totale di fornitori, ma il numero di fornitori per ogni città
- Per estrarre questo tipo di informazioni devo raggruppare le righe secondo l'attributo City

- Esempio: estrarre il numero di fornitori di ogni città con status di almeno 10
 - Attributo discriminante: City
 - Attributo della funzione aggregata: SNum (oppure *)
 - Query:

```
select City, count(SNum)
from S
where Status> = 10
group by City;
```

City	COUNT(SNum)
Athens	1
London	2
Paris	2

- Funzionamento
 - 1. Viene prima eseguita la query come se non ci fosse né la clausola *group by* né la funzione aggregata

select City, SNum
from S
where Status> = 10;

City	SNum
London	S1
Paris	S2
Paris	S3
London	S 4
Athens	S5

- 2. Le righe vengono raggruppate in base ai valori dell'attributo nella clausola *group by*
 - group by City

City	SNum	City	SNı
London	S1	Athens	S5
Paris	S2	London	S1
Paris	S3	London	S4
London	S 4	Paris	S2
Athens	S5	Paris	S3

- 3. Viene ora considerata la clausola select e la funzione aggregata viene calcolata sull'attributo *SNum* per ogni gruppo nella tabella
 - select City, count(SNum)

	City
	Athens
	London
	Paris
	Athens London

Cosa succede se considero un attributo diverso da City nella select?

 select SName, count(SNum) from S where Status> = 10 group by City;

City	SName	SNum
Athens	Adams	S5
London	Smith	S1
London	Clark	S4
Paris	Jones	S2
Paris	Blake	S3



SName	COUNT(SNum)
Adams	1
?	2
?	2

• La sintassi SQL-92 è:

```
select SottinsiemeAttributiDiscriminanti,
FunzioneAggregata(AltroAttributo)
from Tabelle
where Condizione
group by AttributiDiscriminanti;
```

- La target list può comprendere esclusivamente attributi discriminanti e funzioni aggregate su attributi non discriminanti
- Dato che ogni gruppo ha lo stesso valore per gli attributi discriminanti, la funzione aggregata, che calcola un solo valore per ogni gruppo, non dà problemi di differenza di cardinalità
- Una query con funzioni aggregate che non rispetta questa sintassi viene rifiutata dal parser SQL indipendentemente dalle cardinalità effettive
- In altre parole, si deve considerare la sintassi dell'intero costrutto e non si può semplicemente aggiungere la clausola group by a una select espressa in qualsiasi forma

- Esempio: la query
 select City, Status, count(SNum)
 from S
 group by City;
 è scorretta!
- Infatti, sebbene count(SNum) restituisca un unico valore, per ogni città si possono avere valori diversi per Status

City	Status	SNum
Athens	30	S5
London	20	S1
London	20	S4
Paris	10	S2
Paris	30	S3



City	Status	COUNT(S Num)
Athens	30	1
London	20	2
Paris	?	2

- È possibile considerare più di un attributo per formare raggruppamenti più fini
- Esempio: estrarre il numero di fornitori per città e status

select City, Status, count(SNum) from S

where Status > = 10 group by City, Status;

City	Status	COUNT(SNum)
Paris	10	1
Paris	30	1
Athens	30	1
London	20	2

23

L'ordine degli attributi nella clausola group by non
è importante (a differenza della order by)

Passi di esecuzione

City	Status	SNum
London	20	S1
Paris	10	S2
Paris	30	S3
London	20	S4
Athens	30	S5

from S
where Status> = 10

Status SNum City London 20 **S1 Paris** S2 10 Paris 30 **S**3 S4 London 20 Athens 30 S₅

group by City, Status

City	Status	SNum
Paris	10	S2
Paris	30	S3
Athens	30	S5
London London	20 20	S1 S4

select City,
Status,
count(SNum)

City	Status	count (SNum)
Paris	10	1
Paris	30	1
Athens	30	1
London	20	²⁵ 2

 select Attr1, Attr2, sum(Attr3) from Tab group by Attr1, Attr2;

Attr1	Attr2	Attr3
Α	X	2
Α	X	3
Α	Υ	3
Α	Υ	7
В	Χ	4
В	X	1
В	Z	5
В	Z	2

Attr1	Attr2	Attr3
Α	X	2
Α	Υ	3
Α	Χ	3
Α	Υ	7
В	Z	5
В	Χ	4
В	Z	2
В	X	1

Attr1	Attr2	Attr3
Α	Χ	2
Α	Υ	3
В	Z	5
Α	Χ	3
В	Χ	4
Α	Υ	7
В	Z	2
В	X	1

Attr1	Attr2	Attr3
A A	X	2 3
A	Y	3
A	Y	7
B	X	4
B	X	1
B	Z	5
B	Z	2

Attr1	Attr2	SUM(Attr3)
Α	X	5
Α	Υ	10
В	X	5
В	Z	7

Nota: se Attr1, Attr2 contengono NULL viene creato un gruppo separato con le tuple che contengono NULL

- Nota: SQL-99 attenua la restrizione sulla sintassi della select con group by rendendo possibile includere nella target list, al di fuori delle funzioni aggregate, anche attributi funzionalmente dipendenti da quelli discriminanti
- Quindi in SQL-99 è possibile scrivere
 select PName, sum(Qty)
 from SP join P on SP.Pnum=P.Pnum
 group by P.PNum;
 per ricavare il nome e il numero totale di parti fornite
- PostgreSQL supporta la sintassi SQL-99 permettendo, come nell'esempio, di avere nella target list, al di fuori delle funzioni aggregate, attributi la cui chiave primaria è tra gli attributi discriminanti, Oracle supporta esclusivamente SQL-92, MySQL supporta SQL-99 solo recentemente (da 5.7.5)

- È possibile specificare condizioni che devono valere, anziché sulle righe, su gruppi di righe
- Caso 1: estrarre il numero di fornitori per città con status almeno uguale a 20
- Caso 2: elencare le città con almeno due fornitori

- Caso 1: estrarre il numero di fornitori per città con status di almeno 20
- Si può controllare la condizione riga per riga, quindi si usa una semplice where

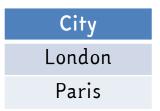
select City, count(SNum)
from S
where Status > = 20
group by City;

City	COUNT(SNum)
Athens	1
London	2
Paris	1

- Come risolvo il Caso 2?
 - Si può controllare la condizione solo sul risultato della funzione aggregata, non su singole righe!

- · Caso 2: elencare le città con almeno due fornitori
- Si introduce la clausola having

```
select City
from S
group by City
having count(*) > = 2;
```



• La clausola *having* viene verificata al termine del raggruppamento eseguito con la *group by*

- È importante comprendere la differenza tra i predicati della clausola where e quelli della clausola having: nella clausola having generalmente vanno messe solo condizioni in cui compaiono funzioni aggregate
- Esempio: elencare le città con almeno due fornitori con status almeno uguale a 20

 Esempio: elencare le città con almeno due fornitori con status almeno uguale a 20

```
select City
from S
where Status> = 20
group by City
having count(*) > = 2;
```

City London

Esercizio 4.1a

• Estrarre la quantità totale di parti rosse fornite da ciascun fornitore (mostrare nome del fornitore e quantità totale di parti). Controllare che nel risultato non siano presenti i fornitori che non forniscono parti rosse come Adams, che non fornisce nessuna parte, e Blake, che fornisce solo parti verdi.

SupplierName	TotQTY
Smith	600
Jones	300
Clark	300

Esercizio 4.1b

• Estrarre la quantità totale di parti rosse fornite da ciascun fornitore compresi i fornitori che non forniscono nessuna parte, per i quali dovete mostrare 0 (mostrare nome del fornitore e quantità totale di parti). Nel risultato ci aspettiamo di avere Adams, che non fornisce nessuna parte, ma non Blake, che fornisce solo parti verdi.

SupplierName	TotQTY
Smith	600
Jones	300
Clark	300
Adams	0

Esercizio 4.1c

• Estrarre la quantità totale di parti rosse fornite da ciascun fornitore compresi i fornitori che non forniscono nessuna parte rossa (mostrare nome del fornitore e quantità totale di parti) (suggerimento: sfruttare la condizione del join). Nel risultato ci aspettiamo di avere sia Adams, che non fornisce nessuna parte, che Blake, che fornisce solo parti verdi.

SupplierName	TotQTY
Smith	600
Jones	300
Clark	300
Adams	0
Blake	0

Esercizio 4.2

 Considerando solo forniture di oltre 100 parti, estrarre le città in cui i fornitori, considerati insieme, forniscono in totale almeno 1000 parti (mostrare la città e la quantità totale di parti)

Esercizio 4.3

 Estrarre le città in cui ci sono almeno due fornitori che hanno fornito ognuno almeno due prodotti di diverso colore (suggerimento: scrivere prima la query che estrae le informazioni sulle coppie di parti di diverso colore fornite dallo stesso fornitore)