



**Educazione
Lavoro
Istruzione
Sport**

Database Design e SQL
Christian Finucci

DML OPERATORI AGGREGATI GROUP BY

Data Analytics



OPERATORI AGGREGATI

Gli operatori aggregati eseguono elaborazioni su insiemi di valori e ne restituiscono uno (ad esempio il voto medio degli studenti di un'università)

Per analisi più avanzate può essere necessario effettuare raggruppamenti e POI effettuare calcoli aggregati (ad esempio il voto medio degli studenti diviso per fasce d'età)



Operatori aggregati

Gli operatori aggregati base di SQL sono operatori di

- **Conteggio (*count*)**
- **Minimo (*min*)**
- **Massimo (*max*)**
- **Media (*avg*)**
- **Totale (*sum*)**

Conteggio

Scrivere la query per contare le auto presenti

Scrivere la query per contare le auto della fiat

id	marca	modello	colore	targa	data_acquisto	id_categoria
1	fiat	punto	NULL	AA000AA	NULL	1
2	ford	focus	blu	BB000BB	NULL	1
3	ferrari	GTO	rosso	AA123BB	2022-01-25	4
4	fiat	panda	rosso	PP123AA	2022-01-25	1
5	fiat	panda	verde	CC000CC	2019-10-10	1
7	fiat	multipla	rosa	AA000CC	2019-01-01	3
8	fiat	multipla	oro	AA111AA	2019-10-01	3
10	fiat	freemont	NULL	AA222AA	NULL	2
11	opel	corsa	nero	AA222AB	NULL	1
12	opel	corsa	blu	CC222AB	2022-01-26	1
14	opel	mokka	verde	AA222BB	2019-01-01	2

Conteggio

Scrivere la query per contare le auto della fiat

```
select count(*)  
from auto  
where marca ="fiat";
```

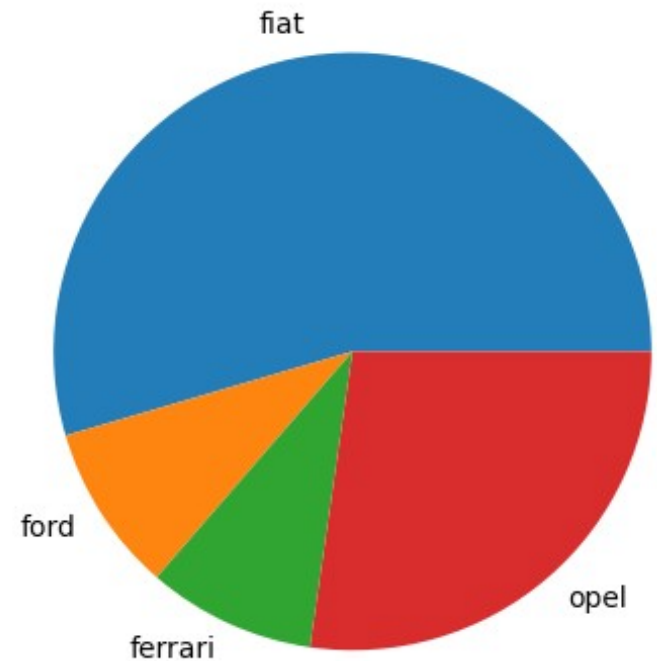
```
+-----+  
| count(*) |  
+-----+  
|         6 |  
+-----+
```

Nella stessa maniera si possono conteggiare le auto delle altre marche.

Conteggio e raggruppamento

Dal punto di vista della data analysis sarebbe interessante ottenere un risultato così strutturato

```
+-----+-----+
| marca  | count(*) |
+-----+-----+
| fiat   |         6 |
| ford   |         1 |
| ferrari|         1 |
| opel   |         3 |
+-----+-----+
```



Conteggio e raggruppamento

Di fatto abbiamo effettuato un' operazione di conteggio (operatore aggregato) ma **raggruppando** i dati per ogni marca presente nella tabella




marca	count (*)
fiat	6
ford	1
ferrari	1
opel	3

raggruppamento

Un resultset così strutturato è ottenibile mediante una query così strutturata

```
select marca, count(*)  
from auto  
group by marca;
```



marca	count (*)
fiat	6
ford	1
ferrari	1
opel	3

Ordinamento

E' possibile introdurre una clausola di ordinamento

```
select marca, count(*)  
from auto  
group by marca  
order by count(*) desc;
```



marca	count(*)
fiat	6
opel	3
ford	1
ferrari	1

Ridenominazione

Con la ridenominazione il resultset risulta sicuramente più facilmente interpretabile

```
select marca, count(*) as numero  
from auto  
group by marca  
order by numero desc;
```




marca	numero
fiat	6
opel	3
ford	1
ferrari	1

Clausola Having

Solo dopo aver raggruppato e contato sarà possibile introdurre una condizione sul valore di aggregazione, ad esempio escludendo le marche con meno di due auto

```
select marca, count(*) as numero  
from auto  
group by marca  
having numero>2  
order by numero desc;
```



+	-	-	-	-	-	+	-	-	-	-	-	+
	mar	ca		nu	me	r	o					
+	-	-	-	-	-	+	-	-	-	-	-	+
	f	i	a	t			6					
	o	p	e	l			3					
+	-	-	-	-	-	+	-	-	-	-	-	+

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Creare il database **test** e popolare la tabella **esami** esattamente come nell'esempio a video.

Hint

```
create table esami(  
    studente varchar(20),  
    materia varchar(20),  
    voto smallint unsigned  
);
```

```
insert into esami values("Rino","canto",30);  
insert into esami values("Nino","analisi 1",20);  
insert into esami values("Rino","analisi 1",30);  
insert into esami values("Nino","fisica 1",30);
```

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Problema:

determinare la media dei voti *studente per studente*

- 1) Di Nino
- 2) Di Rino

determinare la media dei voti materia per materia

- 3) Di analisi 1
- 4) Di canto
- 5) Di fisica 1

Con gli operatori aggregati posso calcolare la media studente per studente, eseguendo tante query separate

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Alle query è possibile aggiungere una clausola **GROUP BY** che consente di raggruppare le righe del ResultSet in base ad un certo attributo (ex il nome dello studente).

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

```
select avg(voto) AS "media studente", studente
from esami
group by (studente);
```

media studente	studente
25.0000	Nino
30.0000	Rino

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

```
select avg(voto) AS "media  
studente", studente  
from esami  
group by(studente) ;
```

- 1) Creare un database università
- 2) Creare e popolare la tabella esami
- 3) Scrivere e testare la query per determinare la media dei voti delle varie materie;



Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

```
select avg(voto) AS "media materia",materia  
from esami  
group by(materia);
```

media materia	materia
25.0000	analisi 1
30.0000	canto
30.0000	fisica 1



Group by

Problema: visualizzare la materia e il numero di studenti che ne hanno sostenuto l'esame, ordinando le materie in ordine alfabetico

```
+-----+-----+
| numero_esaminati | materia |
+-----+-----+
|                2 | analisi 1 |
|                1 | canto    |
|                1 | fisica 1 |
+-----+-----+
```



Group by

Problema: visualizzare la materia e il numero di studenti che ne hanno sostenuto l'esame, ordinando le materie in ordine alfabetico

```
select count(*) as numero_esaminati, materia
  from esami
  group by materia
  order by materia;
```

numero_esaminati	materia
2	analisi 1
1	canto
1	fisica 1



Group by

Problema: visualizzare la materia e il numero di studenti che ne hanno sostenuto l'esame, ordinando le materie in ordine alfabetico

```
+-----+-----+
| numero_esaminati | materia |
+-----+-----+
|           1      | canto  |
|           1      | fisica 1 |
|           2      | analisi 1 |
+-----+-----+
```



Group by

Problema: visualizzare la materia e il numero di studenti che ne hanno sostenuto l'esame, ordinando in base al numero di studenti esaminati

```
select count(studente) AS numero_esaminati, materia
from esami
group by materia
order by numero_esaminati;
```

numero_esaminati	materia
1	canto
1	fisica 1
2	analisi 1



Operatore group by, clausola having

Problema: materie che hanno la media superiore a 28

NOTA: Per imporre condizioni sull'operatore aggregato si utilizza la clausola HAVING (l'utilizzo della clausola WHERE è riservato alla selezione di righe)
HAVING va posta **dopo** il GROUP BY

media materia	materia
25.0000	analisi 1
30.0000	canto
30.0000	fisica 1



media materia	materia
30.0000	canto
30.0000	fisica 1

Problema: materie che hanno la media superiore a 28

```
mysql> select avg(voto) as "media", materia  
-> from esami  
-> where avg(voto) >28  
-> group by materia;  
ERROR 1111 (HY000): Invalid use of group function  
mysql> 
```

Il where non può essere utilizzato con operatori aggregati quando è presente un raggruppamento: si utilizza la clausola having

Operatore group by, clausola having

Per imporre condizioni sull'operatore aggregato si utilizza la clausola HAVING (l'utilizzo della clausola WHERE è riservato alla selezione di righe)

HAVING va posta **dopo** il GROUP BY

Problema: materie che hanno la media superiore a 28

```
select avg(voto) AS "media materia", materia
from esami
group by(materia)
having avg(voto) >28;
```

media materia	materia
25.0000	analisi 1
30.0000	canto
30.0000	fisica 1



media materia	materia
30.0000	canto
30.0000	fisica 1

FOCUS: GROUP BY, HAVING

REGOLE DI COSTRUZIONE DELLE QUERY ANALITICHE

- 1) Quando nella SELECT sono presenti operatori aggregati (ex SUM, AVG) e proiezioni semplici sulle colonne è **obbligatorio** raggruppare (GROUP BY) in base a queste ultime
- 2) Quando voglio porre condizione sui valori delle colonne semplici continuo ad usare la clausola WHERE
- 3) Quando voglio porre condizione sui valori degli operatori aggregati devo utilizzare la clausola HAVING che va posta subito dopo il GROUP BY

Focus: Ordine keyword

Select

From

[Where]

[Group by]

[Having]

[Order by]

[limit]

Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Ottenere un Resultset con **media dei voti e studente** degli studenti il cui nome inizia con 'N'



Operatore Group by

Ottenere un Resultset con **media dei voti e studente** degli studenti il cui nome inizia con 'N'

```
select avg(voto), studente
from esami
where studente like "N%"
group by studente;
```

+	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+
	avg(voto)		studente															
+	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+
	25.0000		Nino															
+	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+



Operatore Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Ottenere un Resultset con **media dei voti** e **studente** ma solo delle materie che iniziano con la lettera 'A'



Group by

studente	materia	voto
Rino	canto	30
Nino	analisi 1	20
Rino	analisi 1	30
Nino	fisica 1	30

Ottenere un Reultset con **media dei voti** e **studente** ma solo delle materie che iniziano con la lettera 'A'

```
mysql> select avg(voto), studente  
-> from esami  
-> where materia like "a%"  
-> group by studente;
```

avg(voto)	studente
20.0000	Nino
30.0000	Rino

2 rows in set (0.01 sec)



Operatore Group by

Media dei voti e studente ma solo delle materie che iniziano con la lettera 'A'

```
mysql> select avg(voto) as "media", studente from esami where materia like "a%" group by studente;
```

media	studente
20.0000	Nino
30.0000	Rino

```
mysql> select avg(voto) as "media", studente from esami group by studente having materia like "a%";  
ERROR 1054 (42S22): Unknown column 'materia' in 'having clause'
```



Group by

Ottenere un Reultset con **media dei voti** e **studente** ma solo delle materie che contengono un '1' nel nome

```
select avg(voto), studente
from esami
where materia like "%1%"
group by studente;
```

+	-----	+	-----	+
	avg(voto)		studente	
+	-----	+	-----	+
	25.0000		Nino	
	30.0000		Rino	
+	-----	+	-----	+



Riepilogo

- A cosa servono gli operatori aggregati
- Quante righe e quante colonne ha il Resultset di un operatore aggregato
- Come si comporta un operatore aggregato relativamente ai valori null
- Cosa determina
 - ***Select count(*)..***
 - ***Select count(voto)..***



FINANCE

CREAZIONE DELLA BASE DI DATI

1) Creare il database finance e renderlo attivo

2) Creare la tabella vendita

```
create table vendita(  
  id int unsigned primary key auto_increment,  
  regione varchar(20),  
  provincia char(2) not null,  
  data date,  
  incasso smallint  
);
```

IMPORT BULK DEI DATI

3) Importare i dati

```
INSERT INTO `vendita` VALUES (1,'Abruzzo','TE','2020-05-01',8000),
(2,'Abruzzo','PE','2020-05-01',9500),(3,'Lazio','RM','2021-03-01',30100),
(4,'Lazio','LT','2021-06-01',10300),(5,'Liguria','GE','2021-04-01',12000),
(6,'Sardegna','CA','2021-08-01',20000),(7,'Puglia','FG','2022-01-01',11030),
(8,'Sicilia','PA','2019-07-01',21000),(9,'Lombardia','MI','2022-02-
01',22500),(10,'Veneto','VR','2020-07-01',18000),(11,'Umbria','TR','2020-03-
01',9600),(12,'Toscana','FI','2021-12-01',17000),(13,'Piemonte','NO','2021-
09-01',13600),(14,'Campania','NA','2019-08-01',11200),
(15,'Piemonte','TO','2020-03-01',15600),(16,'Molise','IS','2019-06-
01',30000),(17,'Marche','FM','2018-05-01',12000),(18,'Sardegna','SS','2018-
06-01',21000),(19,'Lazio','FR','2021-03-01',9900),(20,'Abruzzo','AQ','2019-
04-01',10000),(21,'Emilia-Romagna','RA','2018-06-01',13000),(22,'Emilia-
Romagna','RN','2018-06-01',15000),(23,'Basilicata','PZ','2019-01-01',12400),
(24,'Calabria','CZ','2020-06-01',20000),(25,'Calabria','CS','2020-07-
01',21000),(26,'Liguria','SV','2021-01-01',9800),(27,'Toscana','PI','2022-
01-01',17000),(28,'Toscana','LU','2022-01-01',14000),(29,'Trentino-Alto
Adige','BZ','2021-02-01',12000),(30,'Trentino-Alto Adige','TN','2021-02-
01',14900),(31,'Valle d\'Aosta','AO','2021-06-01',22000),
(32,'Puglia','BA','2022-01-01',18700),(33,'Lazio','RM','2020-01-31',20000),
(34,'Lazio','RM','2020-02-28',10500);
```

IMPORT BULK DEI DATI

3) Importare i dati

```
INSERT INTO `vendita` VALUES (1, 'Abruzzo', 'TE', '2020-05-01', 8000),
(2, 'Abruzzo', 'PE', '2020-05-01', 9500), (3, 'Lazio', 'RM', '2021-03-01', 30100),
(4, 'Lazio', 'LT', '2021-06-01', 10300), (5, 'Liguria', 'GE', '2021-04-01', 12000),
(6, 'Sardegna', 'CA', '2021-08-01', 20000), (7, 'Puglia', 'FG', '2022-01-01', 11030),
(8, 'Sicilia', 'PA', '2019-07-01', 21000), (9, 'Lombardia', 'MI', '2022-02-
01', 22500), (10, 'Veneto', 'VR', '2020-07-01', 18000), (11, 'Umbria', 'TR', '2020-03-
01', 9600), (12, 'Toscana', 'FI', '2021-12-01', 17000), (13, 'Piemonte', 'NO', '2021-
09-01', 13600), (14, 'Campania', 'NA', '2019-08-01', 11200);

INSERT INTO `vendita` VALUES (15, 'Piemonte', 'TO', '2020-03-01', 15600),
(16, 'Molise', 'IS', '2019-06-01', 30000), (17, 'Marche', 'FM', '2018-05-01', 12000),
(18, 'Sardegna', 'SS', '2018-06-01', 21000), (19, 'Lazio', 'FR', '2021-03-01', 9900),
(20, 'Abruzzo', 'AQ', '2019-04-01', 10000), (21, 'Emilia-Romagna', 'RA', '2018-06-
01', 13000), (22, 'Emilia-Romagna', 'RN', '2018-06-01', 15000),
(23, 'Basilicata', 'PZ', '2019-01-01', 12400), (24, 'Calabria', 'CZ', '2020-06-
01', 20000), (25, 'Calabria', 'CS', '2020-07-01', 21000), (26, 'Liguria', 'SV', '2021-
01-01', 9800), (27, 'Toscana', 'PI', '2022-01-01', 17000),
(28, 'Toscana', 'LU', '2022-01-01', 14000), (29, 'Trentino-Alto Adige', 'BZ', '2021-
02-01', 12000), (30, 'Trentino-Alto Adige', 'TN', '2021-02-01', 14900), (31, 'Valle
d\'Aosta', 'AO', '2021-06-01', 22000), (32, 'Puglia', 'BA', '2022-01-01', 18700),
(33, 'Lazio', 'RM', '2020-01-31', 20000), (34, 'Lazio', 'RM', '2020-02-28', 10500);
```


ANALISI DEI DATI

- 1) Contare il numero totale di vendite
- 2) Contare il numero totale di vendite della toscana
- 3) incasso totale
- 4) incasso totale del lazio
- 5) incasso totale di RM
- 6) incasso totale del 2020
- 7) incasso totale di RM del 2020
- 8) incasso totale per ogni regione
- 9) incasso totale per ogni regione e provincia
- 10) incasso totale per ogni regione e provincia nel 2020
- 11) provincie che hanno incassato più di 15000 (in ordine di incasso)
- 12) regioni che hanno incassato meno di 15000 (in ordine di incasso)
- 13) Determinare le tre regioni che hanno incassato di più (in ordine di incasso)

ANALISI DEI DATI

1) Contare il numero totale di vendite

34

2) Contare il numero totale di vendite della toscana

3

3) incasso totale

532630

4) incasso totale del lazio

80800

5) incasso totale di RM

60600

6) incasso totale del 2020

132200

7) incasso totale di RM del 2020

30500

ANALISI DEI DATI

- 1)incasso delle tre regioni che hanno incassato meno(in ordine crescente di incasso)

```
select sum(incasso) as totale ,regione from vendita  
group by regione order by totale;  
(9600,Umbria) , (11200 Campania) , (12000 ,Marche)
```

- 2)incasso totale per ogni regione e provincia
- 3)incasso totale per ogni regione e provincia nel 2020
- 4)provincie che hanno incassato più di 15000 (in ordine di incasso)
- 5)regioni che hanno incassato meno di 15000 (in ordine di incasso)
- 6)Determinare le tre regioni che hanno incassato di più (in ordine di incasso)