

SQL

3 lezione

operatori aggregati esercizi

Francesca Gasparini
gasparini@disco.unimib.it

SQL e algebra relazionale

date le due relazioni R1(A1,A2) R2(A3,A4)

```
SELECT R1.A1, R2.A4
FROM R1, R2
WHERE R1.A2 = R2.A3
```

$PROJ_{A1,A4} (SEL_{A2=A3} (R1 JOIN R2))$

- prodotto cartesiano (**FROM**)
- selezione (**WHERE**)
- proiezione (**SELECT**)

SQL e algebra relazionale 2

possono essere necessarie ridenominazioni

- nel prodotto cartesiano
- nella target list

```
SELECT X.A1 AS B1, Y.A4 AS B2
FROM R1 X, R2 Y, R1 Z
WHERE X.A2 = Y.A3 AND Y.A4 = Z.A1
```

$PROJ_{A1,A4} (SEL_{A2=A3 \wedge A4=A1} (R1 JOIN R2 JOIN REN_{B1,B2 \leftarrow A1,A4} (R1)))$

Maternità	Madre Figlio		Persone		
			Nome	Età	Reddito
Maternità	Luisa	Maria	Andrea	27	21
	Luisa	Luigi	Aldo	25	15
	Anna	Olga	Maria	55	42
	Anna	Filippo	Anna	50	35
	Maria	Andrea	Filippo	26	30
	Maria	Aldo	Luigi	50	40
Paternità	Padre Figlio		Franco	60	20
	Sergio	Franco	Olga	30	41
	Luigi	Olga	Sergio	85	35
	Luigi	Filippo	Luisa	75	87
	Franco	Andrea			
	Franco	Aldo			

join e join esplicito

Query: I padri di persone che guadagnano più di 20

$PROJ_{Padre} (paternita JOIN_{Figlio=Nome} SEL_{Reddito>20} (persone))$

```
SELECT distinct padre
FROM persone, paternita
WHERE figlio = nome and reddito > 20
```

```
SELECT distinct padre
FROM persone, [INNER] JOIN paternita
ON nome = figlio
WHERE reddito > 20
```

self join

Query: mostrare nome, reddito e reddito del padre delle persone che guadagnano più dei rispettivi padri

$PROJ_{Nome, Reddito, RP} (SEL_{Reddito>RP} (REN_{NP,EP,RP \leftarrow Nome, Eta, Reddito} (persone) JOIN_{NP=Padre} (paternita JOIN_{Figlio=Nome} persone)))$

```
SELECT f.nome, f.reddito, p.reddito
FROM persone p, paternita, persone f
WHERE p.nome = padre and
figlio = f.nome and
f.reddito > p.reddito
```

Self Join con join esplicito

```
SELECT f.nome, f.reddito, p.reddito
FROM persone p, paternita, persone f
WHERE p.nome = padre and
      figlio = f.nome and
      f.reddito > p.reddito
```

```
SELECT f.nome, f.reddito, p.reddito
FROM (persone p JOIN paternita ON p.nome = padre)
JOIN persone f ON figlio = f.nome
WHERE f.reddito > p.reddito
```

SELECT, con ridenominazione del risultato

```
SELECT figlio, f.reddito as reddito,
       p.reddito as redditoPadre
FROM persone p, paternita, persone f
WHERE p.nome = padre and figlio = f.nome
      and f.reddito > p.reddito
```

Join esterno

Query: mostrare padre e madre e nome di ogni figlio

```
SELECT padre, madre, maternita.figlio AS Figlio,
FROM paternita FULL [OUTER] JOIN maternita ON
      paternita.figlio = maternita.figlio
```

Padre	Madre	Figlio
NULL	Luisa	Maria
NULL	Luisa	Luigi
Luigi	Anna	Olga
Luigi	Anna	Filippo
Franco	Maria	Andrea
Franco	Maria	Aldo
Sergio	NULL	Franco

Join esterno 2

Query: mostrare padre e se nota madre e nome di ogni figlio

```
SELECT padre, madre, paternita.figlio
FROM paternita LEFT [OUTER] JOIN maternita
      ON paternita.figlio = maternita.figlio
```

Padre	Madre	Figlio
Luigi	Anna	Olga
Luigi	Anna	Filippo
Franco	Maria	Andrea
Franco	Maria	Aldo
Sergio	NULL	Franco

Operatori aggregazione

Sono un'estensione rispetto all'Algebra Relazionale.

Algebra Relazionale (e SQL base) valuta condizioni di clausola **WHERE** su tuple individuali (controlla tupla per tupla se condizioni verificate).

Gli operatori aggregati operano su gruppi di tuple di relazione per:

Effettuare calcoli sull'insieme di tuple;
Verificare condizioni relative all'insieme di tuple.

Operatori aggregazione

COUNT: conta numero di tuple di tabella
SUM: somma valori o espressioni di attributi
MAX: valore massimo di un attributo di tabella
MIN: valore minimo di un attributo di tabella
AVG: valore medio di un attributo di tabella

GROUP BY: clausola per raggruppamento delle tuple in sottogruppi gestiti come estensione delle normali interrogazioni

Come vengono applicati?

- 1) Si esegue l'interrogazione sulla base di clausole FROM e WHERE.
- 2) Si applica l'operatore aggregato alla tabella risultato dell'interrogazione.

Operatori aggregazione: COUNT

COUNT (< * | [DISTINCT | ALL] AttrEspr >)

Restituisce il numero di valori degli attributi in *ListaAttr*:

DISTINCT: numero di valori distinti diversi da NULL di *AttrEspr*

ALL: numero di valori diversi da NULL di *AttrEspr*

Se non è specificato nulla, **ALL** è l'opzione di default.

COUNT (*) conta numero di righe di una tabella

Operatori aggregazione: COUNT

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
1	Mario	Rossi	Amministrazione	10	45		6 ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Landi	Direzione	7	73		1	Genova	
7	Paola	Bononi	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

Operatori aggregazione: COUNT

Impiegato
Dip

Nome	Indirizzo	Città
------	-----------	-------

Query: Estrarre il numero di impiegati del dipartimento Produzione

```
SELECT COUNT(*) AS impiegati_produzione
FROM Impiegato
WHERE dipart='Produzione'
```

impiegati_produzione
2

Operatori aggregazione: COUNT

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
1	Mario	Rossi	Amministrazione	10	45		6 ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Landi	Direzione	7	73		1	Genova	
7	Paola	Bononi	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

L'operatore aggregato **COUNT** viene applicato al risultato dell'interrogazione:

```
SELECT COUNT(*) AS impiegati_produzione
FROM Impiegato
WHERE dipart='Produzione'
```

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	5
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9

Operatori aggregazione: COUNT

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
1	Mario	Rossi	Amministrazione	10	45		6 ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Landi	Direzione	7	73		1	Genova	
7	Paola	Bononi	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

Query: Determinare il numero di valori di Stipendio nella tabella Impiegato.

```
SELECT COUNT(Stipendio) AS stipendio_tutti
FROM Impiegato
```

stipendio_tutti
9

Operatori aggregazione: COUNT

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
1	Mario	Rossi	Amministrazione	10	45		6 ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Landi	Direzione	7	73		1	Genova	
7	Paola	Bononi	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

Query: Determinare il numero di valori distinti di Stipendio nella tabella Impiegato.

```
SELECT COUNT( DISTINCT Stipendio) AS stipendio_distinct
FROM Impiegato
```

stipendio_distinct
7

Operatori aggregati: COUNT

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Query: Determinare il numero di valori distinti di Stipendio tra gli impiegati del dipartimento Produzione.

```
SELECT COUNT( DISTINCT Stipendio) AS Classi
FROM Impiegato
WHERE Dipart='Produzione'
```

Classi
2

Operatori aggregati: COUNT

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Query: Il numero di figli di Franco

```
SELECT COUNT(*) AS NumFigliDiFranco
FROM Paternita
WHERE Padre = 'Franco'
```

NumFigliDiFranco
2

COUNT e valori nulli

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	35
Maria	55	NULL
Anna	50	35

```
SELECT COUNT(*) FROM persone
```

4

```
SELECT COUNT(reddito) FROM persone
```

3

```
SELECT COUNT(DISTINCT reddito) FROM persone
```

2

Altri operatori di aggregazione

< SUM | MAX | MIN | AVG > ([DISTINCT | ALL] AttrExpr)

Valutano espressioni analizzando tuple del risultato della query che li contiene.

Operano su tuple di tabella prese come insieme di elementi (non su elementi individuali).

Le opzioni **DISTINCT** e **ALL** si applicano come per il caso di **COUNT**.

Altri operatori di aggregazione : SUM

Restituisce la somma dei valori di *AttrExpr* relativamente alle tuple di tabella specificate nella query. Opera su attributi numerici o su intervalli di tempo.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma

Query: Determinare la somma degli stipendi degli impiegati afferenti al dipartimento di Amministrazione.

```
SELECT SUM(Stipendio)
FROM Impiegato
WHERE Dipart='Amministrazione'
```

Sum(Stipendio)
85

Altri operatori di aggregazione : MAX MIN

Restituisce il massimo (minimo) dei valori di *AttrExpr* relativamente alle tuple di tabella, specificate nella query. Opera su attributi ordinabili (stringhe, ...).

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma

Query: Determinare lo stipendio minimo degli impiegati afferenti al dipartimento di Amministrazione.

```
SELECT MIN(Stipendio) AS MinS
FROM Impiegato
WHERE Dipart='Amministrazione'
```

MinS
40

Altri operatori di aggregazione : AVG

Restituisce la media dei valori di *AttrExpr* relativamente alle tuple di tabella specificate nella query. Opera su valori numerici.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Query: Determinare la media degli stipendi mensili degli impiegati del dipartimento di Amministrazione.

```
SELECT AVG(Stipendio/12) AS StipendioMedio
FROM Impiegato
WHERE Dipart='Amministrazione'
```

StipendioMedio
3,47222222222222

Altri operatori di aggregazione

Query: Determinare lo stipendio massimo quello minimo e quello medio degli impiegati del dipartimento di Amministrazione.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

```
SELECT MAX(Stipendio), MIN(Stipendio), AVG(Stipendio)
FROM Impiegato
WHERE Dipart='Amministrazione'
```

Altri operatori di aggregazione

Dip			Impiegato					
Nome	Indirizzo	Città	Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Amministrazione	Via Tito Livio, 27	Milano	Mario	Rossi	Amministrazione	10	45	Milano
Produzione	P.le Lavater, 3	Torino	Carlo	Bianchi	Produzione	20	36	Torino
Distribuzione	Via Segre, 9	Roma	Giuseppe	Verdi	Amministrazione	20	40	Roma
Direzione	Via Tito Livio, 27	Milano						

Query: Determinare il massimo stipendio tra quelli degli impiegati che lavorano in un dipart con sede a Milano.

```
SELECT MAX(Stipendio)
FROM Impiegato JOIN Dip ON Dipart=Dip.Nome
WHERE Dip.Città='Milano'
```

45

Operatori di aggregazione e Target List

SQL impedisce di includere in una stessa target list funzioni aggregate e espressioni al livello di riga.

Query: Selezionare nome, cognome e stipendio dell'impiegato che ha lo stipendio piu' alto.

```
SELECT Cognome, Nome, MAX(Stipendio)
FROM Impiegato
WHERE Dipart='Amministrazione'
```

Gli operatori di aggregazione NON fanno una selezione; sono funzioni che restituiscono un valore quando applicate ad un insieme.

E' necessario ricorrere a un'altra soluzione (**GROUP BY**).

Operatori di raggruppamento : GROUP BY

La clausola **GROUP BY** serve a definire gruppi omogenei di tuple, specificando una o più colonne (di raggruppamento) sulla base della/e quale/i le tuple sono raggruppate per valori uguali.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Query: Estrarre la somma degli stipendi degli impiegati che afferiscono allo stesso dipartimento.

Operatori di raggruppamento : GROUP BY

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Query: Estrarre la somma degli stipendi degli impiegati che afferiscono allo stesso dipartimento.

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

Come viene eseguita questa query?

Operatori di raggruppamento : GROUP BY (1)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

- 1) Esecuzione query senza considerare GROUP BY e operatori aggregati:

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

Dipart	Stipendio
Amministrazione	45
Produzione	36
Amministrazione	40
Distribuzione	45
Direzione	80
Direzione	73
Amministrazione	40
Produzione	46

Operatori di raggruppamento : GROUP BY (2)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

- 2) Raggruppamento delle righe del risultato come specificato da GROUP BY (per valore di Dipart):

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

Dipart	Stipendio
Amministrazione	45
Amministrazione	40
Produzione	36
Produzione	46
Distribuzione	45
Direzione	80
Direzione	73

Operatori di raggruppamento : GROUP BY (3)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

- 3) Applicazione dell'operatore di aggregazione SUM su Stipendio ai gruppi di righe precedentemente individuati:

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

Dipart	sum(Stipendio)
Amministrazione	125
Produzione	82
Distribuzione	45
Direzione	153

GROUP BY : Esempi

Paternità	Padre	Figlio
	Sergio	Franco
	Luigi	Olga
	Luigi	Filippo
	Franco	Andrea
	Franco	Aldo

Query: Il numero dei figli di ciascun padre.

```
SELECT padre, COUNT(*) AS NumFigli
FROM paternita
GROUP BY Padre
```

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

GROUP BY : Esempi

Dip

Nome	Indirizzo	Città
Amministrazione	Via Tito Livio 27	Milano
direzione	Via Tito Livio 27	Milano
distribuzione	Via Segre 9	Roma
produzione	P.le Lavater 3	Torino
ricerca	Via Morone 6	Milano

Query: Per ogni dipart, lo stipendio medio nei dipartimenti di Milano.

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Id capo
1	Mario	Rossi	Amministrazione	10	45			Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	8
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1 segretario	Genova	
7	Paola	Borroni	Amministrazione	75	40		11 segretario	Venezia	
8	Marco	Franco	Produzione	20	46		4 ingegnere	Roma	9
9	Francesca	Rossi	Vendite	60	60		3 ricercatore	Milano	8
10	Giuseppe	Verdi	Amministrazione	20	40		3 segretario	Genova	

GROUP BY : Esempi

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Id capo
--------------	------	---------	--------	---------	-----------	------------	----------	-------	---------

Dip

Nome	Indirizzo	Città
------	-----------	-------

Query: Per ogni dipart, lo stipendio medio nei dipartimenti di Milano.

```
SELECT I.dipart, AVG(I.Stipendio) AS AvgStip
FROM Impiegato I, dip S
WHERE (I.dipart = S.nome) AND S.Città = 'Milano'
GROUP BY I.dipart
```

dipart	AvgStip
Amministrazione	41,666666667
Direzione	76,5

GROUP BY : Esempi

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
Dip	Nome	Indirizzo	Città						

Query: Per ogni mansione, lo stipendio medio nei dipartimenti di Milano.

```
SELECT I.mansione, AVG(I.Stipendio) AS AvgStip
FROM Impiegato I, dip S
WHERE (I.dipart = S.nome) AND S.Città = 'Milano'
GROUP BY I.mansione
```

mansione	AvgStip
	73
amministrativo	40
dirigente	80
Ingegnere	45
segretaria	40

GROUP BY : Esempi

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
Dip	Nome	Indirizzo	Città						

Query: Per ogni dipart e mansione, lo stipendio medio nei dipartimenti di Milano.

```
SELECT I.dipart, I.mansione AVG(I.Stipendio) AS AvgStip
FROM Impiegato I, dip S
WHERE (I.dipart = S.nome) AND S.Città = 'Milano'
GROUP BY I.dipart, I.mansione
```

dipart	mansione	AvgStip
Amministrazione	amministrativo	40
Amministrazione	Ingegnere	45
Amministrazione	segretaria	40
Direzione		73
Direzione	dirigente	80

Raggruppamenti e Target List

Importante restrizione: una clausola di proiezione di una query contenente la clausola GROUP BY può solo includere:

- 1) Una o più colonne tra le colonne che compaiono nella clausola GROUP BY;
- 2) Operatori di aggregazione (COUNT, SUM, AVG, ...)

Un'interrogazione scorretta: Un'interrogazione corretta:

```
SELECT nome, cognome
FROM persone
GROUP BY cognome
```

```
SELECT cognome, AVG(Età)
FROM persone
GROUP BY cognome
```

Raggruppamenti e Target List

```
SELECT nome, cognome
FROM persone
GROUP BY cognome
```

è scorretta perché a ogni valore dell'attributo cognome possono corrispondere più valori dell'attributo nome. Il risultato finale dell'interrogazione invece deve corrispondere ad una riga per ogni insieme di righe raggruppate dalla clausola GROUP BY.

Raggruppamenti e Target List

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

```
SELECT Ufficio
FROM Impiegato
GROUP BY Dipart
```

```
SELECT Dipart, COUNT(*), Dip.Città
FROM Impiegato AS I JOIN Dip ON I.Dipart=Dip.Nome
GROUP BY Dipart
```

```
SELECT Dipart, COUNT(*), Dip.Città
FROM Impiegato AS I JOIN Dip ON I.Dipart=Dip.Nome
GROUP BY Dipart, Dip.Città
```

si risolve con insieme di attributi di raggruppamento ridondante

Condizioni sui Gruppi

Verificano condizioni su gruppi di tuple aggregate con GROUP BY.

Tipi di clausole di selezione:

WHERE: verifica condizioni su tuple individuali
HAVING: verifica condizioni (espressioni booleane) su gruppi di tuple (opera su raggruppamenti)

Le condizioni in clausola HAVING sono verificate dopo la creazione dei sottogruppi di tuple specificati da GROUP BY.

Condizioni sui gruppi

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcage
--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Dip

Nome	Indirizzo	Città
------	-----------	-------

Query: Trovare i dipartimenti che spendono più di 100 in stipendi.

```
SELECT Dipart, SUM(Stipendio) AS SommaStipendi
FROM Impiegato
GROUP BY Dipart
HAVING SommaStipendi>100
```

Come viene eseguita questa query?

Condizioni sui gruppi (1)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

1) Esecuzione della query senza considerare GROUP BY e operatori aggregati.

```
SELECT Dipart, SUM(Stipendio) AS SommaStipendi
FROM Impiegato
```

GROUP BY Dipart

HAVING SommaStipendi>100

Dipart	Stipendio
Amministrazione	45
Produzione	36
Amministrazione	40
Distribuzione	45
Direzione	80
Direzione	73
Amministrazione	40
Produzione	46

Condizioni sui gruppi (2)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

2) Raggruppamento delle righe del risultato come specificato da GROUP BY.

```
SELECT Dipart, SUM(Stipendio) AS SommaStipendi
FROM Impiegato
GROUP BY Dipart
HAVING SommaStipendi>100
```

Dipart	Stipendio
Amministrazione	45
Amministrazione	40
Amministrazione	40
Produzione	36
Distribuzione	45
Direzione	80
Direzione	73

Condizioni sui gruppi (3)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

3) Applicazione degli operatori di aggregazione SUM, su Stipendio ai gruppi di righe precedentemente costruiti.

```
SELECT Dipart, SUM(Stipendio) AS SommaStipendi
FROM Impiegato
GROUP BY Dipart
HAVING SommaStipendi>100
```

Dipart	SommaStipendi
Amministrazione	125
Produzione	82
Distribuzione	45
Direzione	153

Condizioni sui gruppi (4)

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

4) Selezione dei gruppi risultanti come specificato dalla clausola HAVING.

```
SELECT Dipart, SUM(Stipendio) AS SommaStipendi
FROM Impiegato
GROUP BY Dipart
HAVING SommaStipendi>100
```

Dipart	SommaStipendi
Amministrazione	125
Direzione	153

Condizioni sui gruppi

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30

Query: I padri i cui figli hanno un reddito medio maggiore di 25.

```
SELECT padre, AVG(f.reddito)
FROM persone f JOIN paternita ON figlio = nome
GROUP BY padre
HAVING AVG(f.reddito) > 25
```


WHERE o HAVING?

Specificano entrambe condizioni di selezione:

WHERE: predicati che operano su tuple individuali (selezione di tuple prima di GROUP BY)

HAVING: predicati che operano su gruppi di tuple creati da GROUP BY (selezione raggruppamenti)

Query: Trovare i dipartimenti la cui media degli stipendi degli impiegati che lavorano nell'ufficio 20 è > di 25.

```
SELECT Dipart, AVG(Stipendio)
FROM Impiegati
WHERE Ufficio = 20
GROUP BY Dipart
HAVING AVG(Stipendio) > 25
```

WHERE o HAVING?

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30

Query: I padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 20.

```
SELECT padre, AVG(f.reddito)
FROM persone f JOIN paternita ON figlio = nome
WHERE Età < 30
GROUP BY padre
HAVING AVG(f.reddito) > 20
```

Riassumiamo

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 si vogliono filtrare i gruppi, si ricorre alla clausola **HAVING**, che permette di esprimere condizioni a livello di gruppo (anziché di singola tupla).

Un esempio completo

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premiagred	Mansione	Città	Id capo
Dip	Nome	Indirizzo	Città							

Query: Per ogni dipartimento di Milano in cui il numero di impiegati è almeno 2, si vuole conoscere il valor medio degli stipendi, ordinando il risultato per valori decrescenti di stipendio medio e quindi per dipart.

```
SELECT I.dipart, AVG(Stipendio) AS AvgStipendio
FROM Impiegato AS I, dip AS S
WHERE I.dipart = S.nome AND S.Città = 'Milano'
GROUP BY I.dipart
HAVING COUNT(*) >= 2
ORDER BY AVG(Stipendio) DESC, dipart
```

dipart	AvgStipendio
Direzione	75,5
Amministrazione	41,55555555555555

Riassunto Sintassi SQL

SelectSQL ::=

```
SELECT ListaAttributiOEspressioni
FROM ListaTabelle
[ WHERE CondizioniSemplici ]
[ GROUP BY ListaAttributiDiRaggruppamento ]
[ HAVING CondizioniAggregate ]
[ ORDER BY ListaAttributiDiOrdinamento ]
```

ESERCIZIO

Dato il seguente schema:

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

scrivere le interrogazioni SQL che permettono di determinare:

ESERCIZIO

Query: Le città con un aeroporto di cui non è noto il numero di piste

```
SELECT Città
FROM AEROPORTO
WHERE NumPiste is NULL
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: Le nazioni da cui parte e arriva il volo con codice AZ274

```
SELECT A1.Nazione, A2.Nazione
FROM AEROPORTO AS A1 JOIN VOLO ON
      A1.Città=CittàArr
      JOIN AEROPORTO AS A2 ON CittàPart=A2.Città
WHERE IdVolo= 'AZ274'
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: I tipi di aereo usati nei voli che partono da Torino

```
SELECT TipoAereo
FROM VOLO
WHERE CittàPart='Torino'
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: I tipi di aereo e il corrispondente numero di passeggeri per i tipi di aereo usati nei voli che partono da Torino. Se la descrizione dell'aereo non è disponibile, visualizzare solamente il tipo;

```
SELECT VOLO.TipoAereo, NumPasseggeri
FROM VOLO LEFT JOIN AEREO
      ON VOLO.TipoAereo=aereo.TipoAereo
WHERE CittàPart= 'Torino'
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: Le città da cui partono voli internazionali

```
SELECT CittàPart
FROM AEROPORTO AS A1 JOIN VOLO ON
      CittàPart=A1.Città
      JOIN AEROPORTO AS A2 ON CittàArr=A2.Città
WHERE A1.Nazione <> A2.Nazione
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: Le città da cui partono voli diretti a Bologna, ordinate alfabeticamente;

```
SELECT CittàPart
FROM VOLO
WHERE CittàArr= 'Bologna'
ORDER BY CittàPart
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: Il numero di voli internazionali che partono il giovedì da Napoli;

```
select count(*)
from VOLO join AEROPORTO on CittàArr=Città
where CittàPart = 'Napoli' and Nazione <> 'Italia' and
GiornoSett= 'Giovedì'
```

AEROPORTO(Città, Nazione, NumPiste)
VOLO(IdVolo, GiornoSett, CittàPart, OraPart,
CittàArr, OraArr, TipoAereo)
AEREO(TipoAereo, NumPasseggeri, QtaMerci)

ESERCIZIO

Query: Il numero di voli internazionali che partono ogni settimana da città italiane (farlo in due modi, facendo comparire o meno nel risultato gli aeroporti senza voli internazionali);

a.

```
select count(*), CittàPart
from AEROPORTO as A1 join VOLO on A1.Città=CittàPart
join AEROPORTO as A2 on CittàArr=A2.Città
where A1.Nazione='Italia' and A2.Nazione <> 'Italia'
group by CittàPart
```

b.

```
select count (CittàArr)
from AEROPORTO as A1 join VOLO on A1.Città=CittàPart
join AEROPORTO as A2 on CittàArr=A2.Città
where A1.Nazione='Italia' and A2.Nazione <> 'Italia'
group by CittàPart
```

ESERCIZIO

Le città francesi da cui partono più di venti voli alla settimana diretti in Italia;

```
select CittàPart
from AEROPORTO as A1 join VOLO on A1.Città=CittàPart
join AEROPORTO as A2 on CittàArr=A2.Città
where A1.Nazione='Francia' and A2.Nazione= 'Italia'
group by CittàPart
Having count(*) >20
```