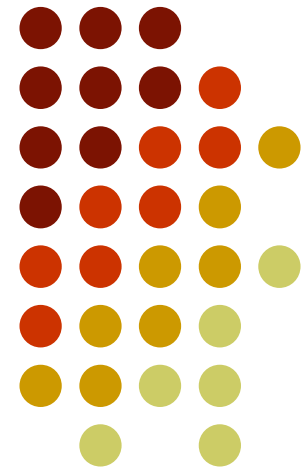


# SQL: Linguaggio di query (II parte)

---

*Elena Ferrari*  
*Basi di Dati*  
*A.A. 2017/2018*





# Operazione di join

- Il join di due relazioni A e B genera tutte le coppie formate da una tupla di A e una tupla di B “semanticamente legate”
- È il metodo che consente di attraversare le associazioni rappresentate mediante il meccanismo delle chiavi esterne

# Join esempio



- Trovare il nome dei corsi e il cognome del docente che li tiene:

idC	corso	CFU	idDoc
53688	Topologia2	6	720
53689	Storia1	12	414
53666	Storia2	18	414

matr	cogn.	login
710	Jones	jones@cs
414	Smith	smith@ee
720	Evams	evams@ee

# Join esempio



Corso. idC	Corso. corso	Corso. CFU	Corso. idDoc	Docente. matr	Docente. cogn	Docente. login
53688	Topologia2	6	720	710	Jones	jones@cs
53688	Topologia2	6	720	414	Smith	smith@ee
53688	Topologia2	6	720	720	Evams	evams@ee
53689	Storia1	12	414	710	Jones	jones@cs
53689	Storia1	12	414	414	Smith	smith@ee
53689	Storia1	12	414	720	Evams	evams@ee
53666	Storia2	18	414	710	Jones	jones@cs
53666	Storia2	18	414	414	Smith	smith@ee
53666	Storia2	18	414	720	Evams	evams@ee



# Prodotto cartesiano in SQL

- Relazioni argomento del prodotto cartesiano vanno nella clausola FROM
  - `SELECT * FROM Corso, Docente;`
- Sintassi alternativa:
  - `SELECT * FROM Corso CROSS JOIN Docente;`



# Operazione di join

- Semantica del join:
  - prodotto Cartesiano di due relazioni al cui risultato è applicata una selezione in base ad un predicato chiamato predicato di join
  - Un predicato di join esprime un'associazione che deve essere verificata dalle tuple risultato dell'interrogazione



# Prodotto cartesiano in SQL

- `SELECT corso, cognome FROM Corso, Docente  
WHERE Corso.idDoc = Docente.matr;`



predicato di join



# Predicato di join

- Siano  $S$  ed  $S'$  le relazioni di cui si vuole effettuare il join
- Sia  $A$  un attributo di  $S$  ed  $A'$  un attributo di  $S'$
- Un **predicato di join per  $S$  ed  $S'$**  è definito come:
  - $A \theta A'$ ,  $\theta$  operatore di SQL
  - o una combinazione booleana di tali predicati





# Operazione di join

- Sintassi base:
  - Le relazioni da congiungere tramite il join vanno nella clausola FROM
  - I predicati di join vanno nella clausola WHERE
  - Viene effettuato un prodotto Cartesiano delle relazioni nella clausola FROM a cui si applica poi la selezione data dal predicato di join

# Operazioni di join



- Lo schema della relazione risultato è dato dalla unione degli schemi delle relazioni operandi
- Il grado della relazione risultato è uguale alla somma dei gradi delle relazioni operandi

# Esempio



- Determinare i titoli dei video noleggiati dai clienti della videoteca

# Esempio



Video

colloc	titolo	regista	tipo
1111	underground	emir kusturica	v
1112	underground	emir kusturica	d
1113	big fish	tim burton	v
1114	big fish	tim burton	d
1115	edward mani di forbice	tim burton	d
1116	nightmare before christmas	tim burton	v
1117	nightmare before christmas	tim burton	d
1118	ed wood	tim burton	d
1119	mars attacks	tim burton	d
1120	il mistero di sleepy hollow	tim burton	d
1121	la sposa cadavere	tim burton	d
1122	la fabbrica di cioccolato	tim burton	d
1123	la fabbrica di cioccolato	tim burton	d
1124	io non ho paura	gabriele salvatores	d
1125	nirvana	gabriele salvatores	d
1126	mediterraneo	gabriele salvatores	d
1127	pulp fiction	quentin tarantino	v
1128	pulp fiction	quentin tarantino	d
1129	le iene	quentin tarantino	d

Noleggio

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?

# Esempio



```
SELECT titolo  
FROM Video, Noleggio  
WHERE Video.colloc = Noleggio.colloc;
```

# Esempio



- Determinare i titoli dei video noleggiati il 15 Marzo 2006 dal cliente con codice 6635:

# Esempio



- Determinare i titoli dei video noleggiati il 15 Marzo 2006 dal cliente con codice 6635:

```
SELECT titolo
```

```
FROM Video, Noleggio
```

```
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006'
```

```
AND Video.colloc = Noleggio.colloc;
```

Risultato:

```
titolo
-----
big fish
la sposa cadavere
la fabbrica di cioccolato
le iene
```

### Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87







# Selezione, proiezione e join

- I padri di persone che guadagnano più di 20:

```
SELECT DISTINCT Padre  
FROM Persone, Paternita  
WHERE Figlio = Nome AND  
Reddito > 20;
```



# Join: sintassi alternative

- JOIN ON:
  - Sintassi: <nome relazione> JOIN <nome relazione>  
ON <predicato>



# Join: sintassi alternative

- **JOIN USING:**
  - Applicabile quando le due relazioni hanno colonne con lo stesso nome
  - Sintassi: `<nome relazione> JOIN <nome relazione> USING (<lista nomi colonne>)`
  - `<lista nomi colonne>` contiene nomi di colonne in comune tra le due relazioni
  - Viene richiesta l'uguaglianza dei valori delle colonne specificate

# Esempio



Video

colloc	titolo	regista	tipo
1111	underground	emir kusturica	v
1112	underground	emir kusturica	d
1113	big fish	tim burton	v
1114	big fish	tim burton	d
1115	edward mani di forbice	tim burton	d
1116	nightmare before christmas	tim burton	v
1117	nightmare before christmas	tim burton	d
1118	ed wood	tim burton	d
1119	mars attacks	tim burton	d
1120	il mistero di sleepy hollow	tim burton	d
1121	la sposa cadavere	tim burton	d
1122	la fabbrica di cioccolato	tim burton	d
1123	la fabbrica di cioccolato	tim burton	d
1124	io non ho paura	gabriele salvatores	d
1125	nirvana	gabriele salvatores	d
1126	mediterraneo	gabriele salvatores	d
1127	pulp fiction	quentin tarantino	v
1128	pulp fiction	quentin tarantino	d
1129	le iene	quentin tarantino	d

Noleggio

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?

# Esempio: sintassi alternative



```
SELECT titolo  
FROM Video JOIN Noleggio ON Video.colloc = Noleggio.colloc  
WHERE codCli = 6635 AND dataNoI = DATE '15-Mar-2006';
```

```
SELECT titolo FROM Video JOIN Noleggio USING (colloc)  
WHERE codCli = 6635 AND dataNoI = DATE '15-Mar-2006';
```



# Natural Join

- Ulteriore estensione (meno diffusa)
  - Viene richiesta l'uguaglianza dei valori **di tutte le colonne** con lo stesso nome nelle due relazioni
  - Sintassi: `<nome relazione> NATURAL JOIN <nome relazione>`
  - Esempio:
    - `SELECT titolo FROM Video NATURAL JOIN Noleggio`  
`WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';`



# Operazione di join

- Nel caso di NATURAL JOIN e JOIN USING le colonne con lo stesso nome nelle due relazioni vengono incluse un'unica volta nel risultato

### Maternità

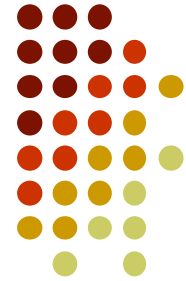
Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87





# Esempio

- Padre e madre di ogni persona





# Esempio

- Padre e madre di ogni persona:

Paternita **NATURAL JOIN** Maternita;

```
SELECT Paternita.Figlio, Padre, Madre  
FROM Maternita, Paternita  
WHERE Paternita.Figlio = Maternita.Figlio;
```

# Esempio



- Padre e madre di ogni persona (altra versioni):

```
SELECT Madre, Paternita.Figlio, Padre  
FROM Maternita JOIN Paternita ON  
    Paternita.Figlio = Maternita.Figlio;
```

```
SELECT *  
FROM Maternita JOIN Paternita USING(Figlio);
```

### Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



# Necessità di ridenominazione

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre



## Persone

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

## Paternità

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

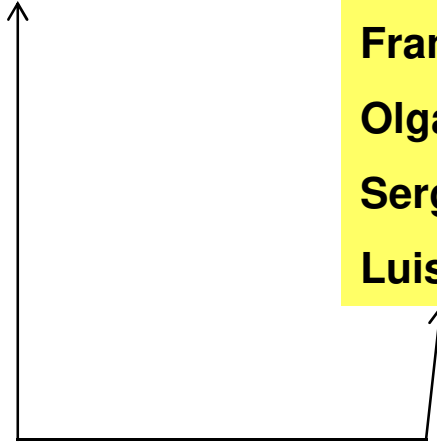
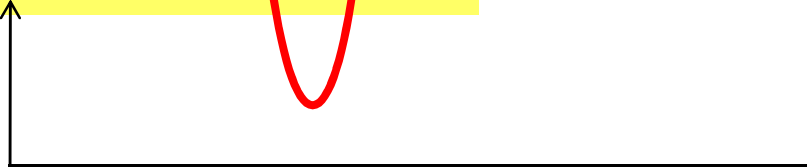
## Persone



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

X

X





# Necessità di ridenominazione

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre:

```
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;
```

p ed f sono alias di relazione

# Esempio

## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00



## Video

colloc	titolo	regista	tipo
1111	underground	emir kusturica	v
1112	underground	emir kusturica	d
1113	big fish	tim burton	v
1114	big fish	tim burton	d
1115	edward mani di forbice	tim burton	d
1116	nightmare before christmas	tim burton	v
1117	nightmare before christmas	tim burton	d
1118	ed wood	tim burton	d
1119	mars attacks	tim burton	d
1120	il mistero di sleepy hollow	tim burton	d
1121	la sposa cadavere	tim burton	d
1122	la fabbrica di cioccolato	tim burton	d
1123	la fabbrica di cioccolato	tim burton	d
1124	io non ho paura	gabriele salvatores	d
1125	nirvana	gabriele salvatores	d
1126	mediterraneo	gabriele salvatores	d
1127	pulp fiction	quentin tarantino	v
1128	pulp fiction	quentin tarantino	d
1129	le iene	quentin tarantino	d

## Noleggio

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?



# Outer Join



- Per ogni video contenente un film di Tim Burton vogliamo visualizzare la sua collocazione, il titolo ed i codici dei clienti che l'hanno **eventualmente** noleggiato:

```
SELECT colloc, titolo, codCli  
FROM Video NATURAL JOIN Noleggio  
WHERE regista = 'tim burton';
```



# Outer join

- In T JOIN S non si ha traccia delle tuple di T che non corrispondono ad alcuna tupla di S
- Questo non è sempre quello che si desidera
- L'operatore OUTER JOIN aggiunge al risultato le tuple di T o S (o entrambe) che non hanno partecipato al join, completandole con NULL
- L'operatore di JOIN originario, per contrasto, è anche detto INNER JOIN



# Outer join

- Esistono diverse varianti dell'outer join:

T OUTER JOIN S:

- **FULL**: sia le tuple di T che quelle di S che non partecipano al join vengono completate ed inserite nel risultato
- **LEFT**: le tuple di T che non partecipano al join vengono completate ed inserite nel risultato
- **RIGHT**: le tuple di S che non partecipano al join vengono completate ed inserite nel risultato



# Outer Join

- Per ogni video contenente un film di Tim Burton vogliamo visualizzare la sua collocazione, il titolo ed i codici dei clienti che l'hanno eventualmente noleggiato

```
SELECT colloc, titolo, codCli  
FROM Film NATURAL JOIN Video  
NATURAL LEFT OUTER JOIN Noleggio  
WHERE regista = 'tim burton';
```

### Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87





# Outer join

- Padre e, se nota, madre di ogni persona

```
SELECT Paternita.Figlio, Padre, Madre  
FROM Paternita LEFT OUTER JOIN Maternita  
ON Paternita.Figlio = Maternita.Figlio;
```



# Funzioni di gruppo

- Nelle espressioni della clausola di proiezione possiamo avere anche espressioni che calcolano valori a partire da insiemi di tuple:
  - Media degli stipendi
  - Massimo voto conseguito



# Funzioni di gruppo

- Le principali funzioni di gruppo previste da SQL sono:
  - **MAX**: determina il massimo in un insieme di valori
  - **MIN**: determina il minimo in un insieme di valori
  - **SUM**: esegue la somma dei valori in un insieme
  - **AVG**: esegue la media dei valori in un insieme
  - **COUNT**: determina la cardinalità di un insieme

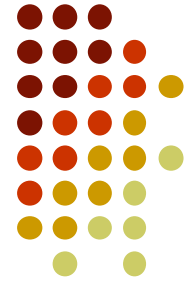




# Funzioni di gruppo

- Operano su un insieme di valori
- Producono come risultato un unico valore (aggregato)
- Sono indicate nella clausola di SELECT
  - Possono essere indicate più funzioni di gruppo contemporaneamente

# Esempio



## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00



# Funzioni di gruppo

```
SELECT MIN(valutaz), AVG(valutaz), MAX(valutaz)  
FROM Film;
```

```
SELECT MIN(valutaz), AVG(valutaz), MAX(valutaz)  
FROM Film  
WHERE genere = 'drammatico';
```



# Funzioni di gruppo

- Ad eccezione di COUNT, sono applicate su insiemi di valori semplici e non su insiemi di tuple
- L'insieme di valori è denotato nel caso più semplice da un nome di colonna, in generale da un'espressione contenente nomi di colonna
- Le funzioni SUM e AVG sono definite solo per insiemi di valori numerici



# Funzione COUNT

- Può avere come argomento anche il carattere speciale '\*' nel qual caso la funzione restituisce il numero di tuple
  - `SELECT COUNT(*)`  
`FROM Persone;`



# Funzioni di gruppo

- Tutte le funzioni di gruppo possono essere usate con il qualificatore **DISTINCT**:
  - eventuali valori duplicati sono eliminati prima di applicare la funzione
  - l'eliminazione dei duplicati è significativa solo per le funzioni **SUM**, **AVG** e **COUNT**
- I valori nulli vengono eliminati prima del calcolo della funzione di gruppo



# Funzioni di gruppo

- Se l'insieme dei valori cui applicare la funzione di gruppo è vuoto:
  - La funzione COUNT restituisce il valore numerico 0
  - Le altre funzioni restituiscono il valore NULL
- Le funzioni di gruppo possono a loro volta essere utilizzate in espressioni aritmetiche:  
es. `SUM(valutaz)/COUNT(*)`

# COUNT DISTINCT



```
SELECT COUNT(Reddito) FROM Persone;
```

```
SELECT COUNT(DISTINCT Reddito) FROM Persone;
```



### Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87





# Funzioni di gruppo: COUNT

- Il numero di figli di Franco:

```
SELECT COUNT(*) AS NumFigliDiFranco
```

```
FROM Paternita
```

```
WHERE Padre = 'Franco';
```

- La funzione di gruppo (count) viene applicata al risultato dell'interrogazione:

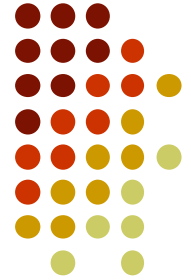
```
SELECT *
```

```
FROM Paternita
```

```
WHERE Padre = 'Franco';
```

**Paternità**

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



**NumFigliDiFranco**

**2**



# Funzioni di gruppo

- Le colonne ottenute dall'applicazione di funzioni di gruppo, come quelle ottenute dall'applicazione di funzioni aritmetiche, sono colonne virtuali, cui è possibile assegnare un nome con la clausola AS



# Funzioni di gruppo

- Le funzioni di gruppo sono valutate solo dopo l'applicazione di tutti i predicati nella clausola WHERE
- Le funzioni di gruppo non possono essere utilizzate nella clausola WHERE (se non in sottointerrogazioni)



# Funzioni di gruppo

- Determinare il titolo e il regista del film drammatico di valutazione minima:

~~SELECT titolo, regista~~

~~FROM Film~~

~~WHERE genere = 'drammatico' AND valutaz = MIN(valutaz);~~

- Avremo bisogno di una sotto-interrogazione (vedi seguito)



# Funzioni di gruppo

- Un'interrogazione scorretta:

```
SELECT Nome, MAX(Reddito)  
FROM Persone;
```

- Di chi sarebbe il nome? La clausola di proiezione deve essere omogenea:

```
SELECT MIN(Età), AVG(Reddito)  
FROM Persone;
```

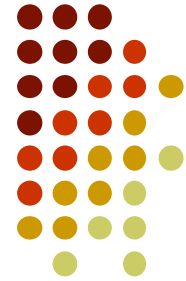


# Raggruppamento

- Le funzioni di gruppo possono essere applicate a partizioni delle relazioni
- Le colonne da usare nel partizionamento sono specificate nella clausola GROUP BY



# Esempio



## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

# Esempio



- Per ogni regista vogliamo determinare quanti suoi film sono presenti in catalogo, di quanti generi diversi e la valutazione minima, media e massima di tali film:

```
SELECT regista,  
    COUNT(*) AS numF,  
    COUNT(DISTINCT genere) AS numG,  
    MIN(valutaz) AS minV,  
    AVG(valutaz) AS avgV,  
    MAX(valutaz) AS maxV  
FROM Film  
GROUP BY regista;
```

avg = media

count = contatore

as <nome colonna  
che voglio creare> =  
crea una nuova  
colonna con quel  
nome contenente l'  
elemento specificato  
prima di as

distinct  
<nome  
colonna> =  
toglie  
eventuali  
doppioni

# Esempio



Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00



# Raggruppamento

- In un'interrogazione contenente una clausola GROUP BY, ogni tupla della relazione risultato rappresenta un gruppo di tuple della relazione su cui l'interrogazione è eseguita
- Nella query precedente i gruppi sono quattro: uno per ogni valore di regista
- Ad ognuno di questi gruppi sono applicate le funzioni di gruppo

# Esempio

- Risultato:

regista	numF	numG	minV	avgV	maxV
emir kusturica	1	1	3.20	3.20	3.20
tim burton	8	5	3.00	3.58	4.00
gabriele salvatores	3	3	3.00	3.43	3.80
quentin tarantino	2	1	3.50	3.75	4.00

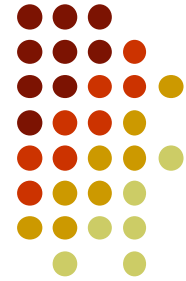


# Provate voi....

- Il numero di figli di ciascun padre:

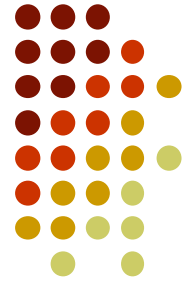


# Esempio



## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00



# Raggruppamento e proiezione

- Se partizioniamo i film per regista, non ha senso richiedere che venga restituito il titolo o la valutazione:

~~SELECT regista, titolo, valutaz~~

FROM Film

GROUP BY regista;

- Dopo aver eseguito il raggruppamento di Film su regista abbiamo quattro tuple:
  - Ognuna corrisponde ad un valore di regista
  - I valori di titolo e valutaz corrispondenti sono tanti!



# Raggruppamento e proiezione



- La clausola di proiezione di un'interrogazione contenente la clausola GROUP BY può solo includere:
  - Una o più colonne tra le colonne che compaiono nella clausola GROUP BY
  - Funzioni di gruppo

### Maternità

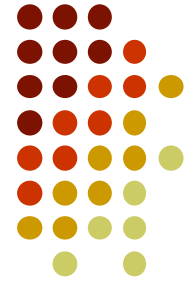
Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



# Raggruppamento



- Per ogni padre, il suo nome, il suo reddito e la media dei redditi dei figli:



# Raggruppamento

- Query scorretta:

```
SELECT Padre, AVG(f.Reddito), p.Reddito  
FROM Persone f JOIN Paternita on Figlio = Nome JOIN  
        Persone p ON Padre =p.Nome  
GROUP BY Padre;
```

- Query corretta:

```
SELECT Padre, AVG(f.Reddito), p.Reddito  
FROM Persone f JOIN Paternita on Figlio = Nome JOIN  
        Persone p ON Padre =p.Nome  
GROUP BY Padre, p.Reddito;
```



# Raggruppamento

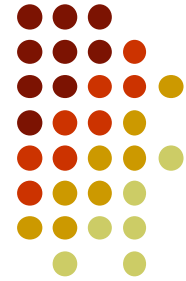
- Più colonne possono essere usate per definire gruppi
- Le funzioni di gruppo possono essere usate anche in presenza di join



# Condizioni sui gruppi

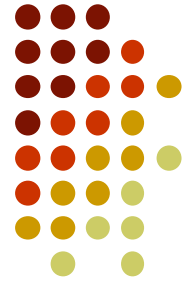
- E' possibile selezionare solo alcuni dei gruppi ottenuti dal partizionamento mediante la clausola **HAVING**
- **HAVING:**
  - E' seguita da un predicato o una combinazione Booleana di predicati
  - Tali predicati possono essere solo predicati che coinvolgono funzioni di gruppo

# Esempio



## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00



# Esempio

- Per ogni regista che ha girato almeno due film prima del 2000, determinare quanti sono tali film, di quanti generi diversi e la valutazione minima, media e massima di tali film:
  - La condizione è definita su valori aggregati
  - Non è possibile usare la clausola WHERE



# Esempio



- Per ogni regista che ha girato almeno due film prima del 2000, determinare quanti sono tali film, di quanti generi diversi e la valutazione minima, media e massima di tali film:

```
SELECT regista, COUNT(*) AS numF,  
        COUNT(DISTINCT genere) AS numG,  
        MIN(valutaz) AS minV, AVG(valutaz) AS avgV,  
        MAX(valutaz) AS maxV  
FROM Film  
WHERE anno < 2000  
GROUP BY regista  
HAVING COUNT(*) >= 2;
```

having count  
controlla se  
ci sono  
almeno/ al  
massimo o  
uguale n  
elementi che  
soddisfano,  
se non ci  
sono annulla  
quel regista

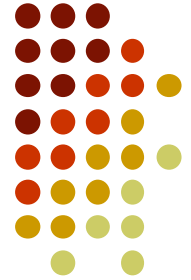
# Esempio



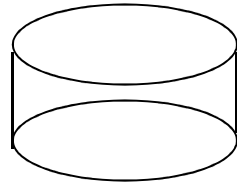
- Risultato:

regista	numF	numG	minV	avgV	maxV
tim burton	5	5	3.00	3.62	4.00
gabriele salvatores	2	2	3.00	3.40	3.80
quentin tarantino	2	1	3.50	3.75	4.00

# Raggruppamento

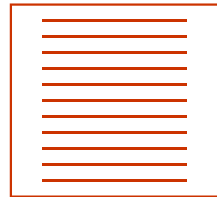


base di dati



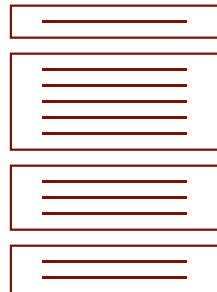
1. ↓

film girati  
prima del 2000



2. ↓

film girati  
prima del 2000  
raggruppati  
per regista

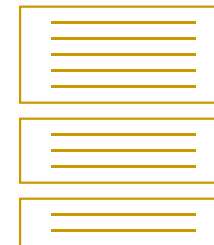


3. →



risultato

4. ↑



film girati prima del 2000  
raggruppati per regista,  
considerando solo registi  
che ne hanno girato  
almeno due



### Maternita

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

### Paternita

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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



# WHERE o HAVING?

- I padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25:

```
SELECT Padre  
FROM Persone JOIN Paternita ON Figlio = Nome  
WHERE Eta < 30  
GROUP BY Padre  
HAVING AVG(Reddito) > 25;
```

reddito medio



# Esempio

- I padri che hanno almeno due figli:

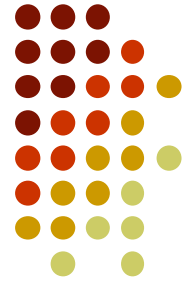
```
SELECT Padre  
FROM Paternita  
GROUP BY Padre  
HAVING COUNT(*) >= 2;
```



# Alternativa senza COUNT

- I padri che hanno almeno due figli:

```
SELECT Padre  
FROM Paternita p JOIN Paternita x ON (p.Padre =  
x.Padre)  
WHERE p.Figlio <> x.Figlio;
```



# Operazioni insiemistiche

- La select da sola non consente di fare unioni, intersezioni, differenze
- SQL prevede dei costrutti ad-hoc:
  - UNION
  - MINUS/EXCEPT
  - INTERSECT





# UNION

- L'operatore insiemistico di unione:

**A UNION B**

- Esegue l'unione delle due espressioni relazionali A e B
  - A e B possono essere generate da istruzioni SELECT
  - richiede la compatibilità di schema tra A e B
  - rimozione dei duplicati
    - UNION rimuove i duplicati
    - UNION ALL non rimuove i duplicati

# Esempio

- Determinare i nomi dei genitori





## Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

## 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
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



## Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

```
SELECT Madre  
FROM Maternita;
```

Madre
Luisa
Luisa
Anna
Anna
Maria
Maria

## Paternità

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

```
SELECT Padre  
FROM Paternita;
```

Padre
Sergio
Luigi
Luigi
Franco
Franco

# Unione

- Determinare i nomi dei genitori:

```
SELECT Padre  
FROM Paternita  
UNION  
SELECT Madre  
FROM Maternita;
```

Padre	Madre	Padre
Sergio	Luisa	Sergio
Luigi	Luisa	Luigi
Luigi	Anna	Franco
Franco	Anna	Luisa
Franco	Maria	Anna
	Maria	Maria

avessi fatto union all allora avevo i doppioni

# Unione



```
SELECT Padre AS Genitore
```

```
FROM Paternita
```

```
UNION
```

```
SELECT Madre AS Genitore
```

```
FROM Maternita;
```

creo la colonna di  
nome genitore dove all  
inizio inserisco tutti i  
padri, poi tutte le  
maddri

**Genitore**

**Sergio**

**Luigi**

**Franco**

**Luisa**

**Anna**

**Maria**



# Notazione posizionale

SELECT Padre, Figlio  
FROM Paternita  
UNION  
SELECT Figlio, Madre  
FROM Maternita;

NO!

SELECT Padre, Figlio  
FROM Paternita  
UNION  
SELECT Madre, Figlio  
FROM Maternita;

OK

figlio se è il secondo  
elemento di select  
allora deve essere il  
secondo anche nel  
altro select dopo  
union

# Notazione posizionale



- Anche con le ridenominazioni non cambia niente:

```
SELECT Padre AS Genitore, Figlio  
FROM Paternita
```

```
UNION
```

```
SELECT Figlio, Madre AS Genitore  
FROM Maternita;
```

- Corretta:

```
SELECT Padre AS Genitore, Figlio  
FROM Paternita
```

```
UNION
```

```
SELECT Madre AS Genitore, Figlio  
FROM Maternita;
```

figlio se è il secondo  
elemento di select  
allora deve essere il  
secondo anche nel  
altro select dopo  
union





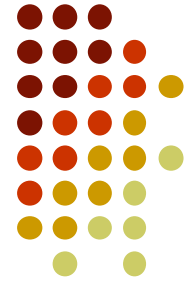
# INTERSECT

- L'operatore insiemistico di intersezione:

**A INTERSECT B**

- Esegue l'intersezione delle due espressioni relazionali A e B
  - A e B possono essere generate da istruzioni SELECT
  - richiede la compatibilità di schema tra A e B

# Esempio



## Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

# Esempio



- Determinare gli anni in cui sono usciti sia film di Tim Burton sia film di Quentin Tarantino:

# ESempio



```
SELECT anno FROM Film WHERE regista = 'tim  
burton'
```

```
INTERSECT
```

```
SELECT anno FROM Film WHERE regista = 'quentin  
tarantino';
```



# Equivalenza

- L'operazione di intersezione può essere eseguita anche mediante:
  - il join
  - l'operatore IN (lo vedremo in seguito servono le sottoquery)



# Esempio

- Determinare gli anni in cui sono usciti sia film di Tim Burton sia film di Quentin Tarantino:

```
SELECT anno
```

```
FROM Film F1, Film F2
```

```
WHERE F1.regista = 'tim burton' AND F2.regista =  
'quentin tarantino' AND F1.anno = F2.anno;
```

versione alternativa piu lunga



# DIFFERENZA

- L'operatore insiemistico di intersezione:

**A EXCEPT/MINUS B**

- Sottrae l'espressione relazionale B all'espressione relazionale A
  - A e B possono essere generate da istruzioni SELECT
  - richiede la compatibilità di schema tra A e B



# Esempio

- determinare gli anni in cui sono usciti film di Tim Burton ma non film di Quentin Tarantino:

```
SELECT anno FROM Film WHERE regista = 'tim  
burton'
```

```
EXCEPT
```

```
SELECT anno FROM Film WHERE regista = 'quentin  
tarantino';
```



# Esempio



- Risultato:

<u>anno</u>
1990
1993
1996
1999
2003
2005
2005



# Equivalenza

- L'operazione di differenza può essere eseguita anche mediante:
  - l'operatore NOT IN (lo vedremo in seguito servono le sottoquery)