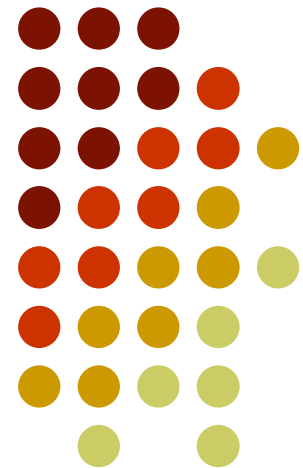
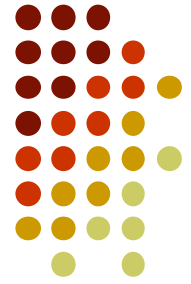


# SQL: sotto-interrogazioni

---

*Elena Ferrari*  
*Basi di Dati*  
*A.A. 2020/2021*





# Sotto-interrogazioni

- Una sotto-interrogazione (subquery) è un'istruzione SELECT contenuta all'interno di un'altra interrogazione
  - permette di suddividere un problema complesso in sotto-problemi più semplici
- È possibile introdurre istruzioni SELECT:
  - in un predicato nella clausola WHERE
  - in un predicato nella clausola HAVING
  - nella clausola FROM (non lo vedremo)

# Sotto-interrogazioni nella clausola WHERE



- La sotto-interrogazione viene usata per:
  - determinare uno o più valori da usare come valori di confronto in un predicato dell'interrogazione esterna
  - esprimere quantificazioni universali/esistenziali



# Sotto-interrogazioni

- Tre tipi:
  - **subquery scalare**: è un comando SELECT che restituisce un solo valore.
    - `SELECT Max(valutazione) FROM Film;`
  - **subquery di colonna**: è un comando SELECT che restituisce una colonna
    - `SELECT valutazione FROM Film;`
  - **subquery di tabella**: è un comando SELECT che restituisce una tabella con più di un attributo:
    - `SELECT titolo, regista FROM Film;`

# 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



# Subquery scalare

- Determinare il titolo di tutti i film che hanno la stessa valutazione del film ``Le iene''
- La formulazione mediante subquery consente di separare il problema in due sottoproblemi
  - valutazione del film Le iene
  - titolo dei film con la stessa valutazione



# Subquery scalare

Determinare il titolo di tutti i film che hanno la stessa valutazione del film ``Le iene":

```
SELECT titolo FROM Film
```

```
WHERE valutaz = (SELECT valutaz FROM Film  
                 WHERE titolo = 'le iene');
```

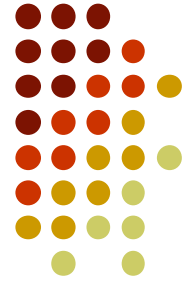


# Formulazione equivalente

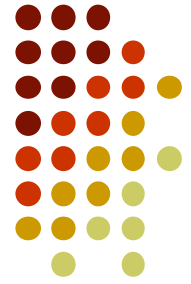
- La formulazione equivalente con il join è caratterizzata da:
  - clausola **FROM** contenente le tabelle referenziate nelle **FROM** di tutte le **SELECT**
  - opportune condizioni di join nella clausola **WHERE** o nel **FROM**
  - eventuali predicati di selezione aggiunti nella clausola **WHERE**



# Formulazione equivalente



```
SELECT titolo FROM Film
WHERE valutaz = (SELECT valutaz
                  FROM Film
                  WHERE titolo = 'le iene');
```



# Formulazione equivalente

```
SELECT titolo FROM Film
WHERE valutaz = (SELECT valutaz
                  FROM Film
                  WHERE titolo = 'le iene');
```

The diagram highlights two instances of the 'Film' table name. The first 'Film' in the FROM clause is circled in yellow and labeled 'F1' with an arrow. The second 'Film' in the subquery's FROM clause is also circled in yellow and labeled 'F2' with an arrow.



# Formulazione equivalente

- Determinare il titolo di tutti i film che hanno la stessa valutazione del film ``Le iene":

```
SELECT titolo FROM Film F1, Film F2  
WHERE ?
```



# Formulazione equivalente

- Determinare il titolo di tutti i film che hanno la stessa valutazione del film ``Le iene'':

```
SELECT titolo FROM Film F1, Film F2
WHERE F1. titolo <> F2. titolo AND F1.valutaz =
      F2.valutaz AND F2.titolo = 'le iene';
```

# Esempio

F1



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

F2

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



# Formulazione equivalente

- Determinare il titolo di tutti i film che hanno la stessa valutazione del film ``Le iene":

```
SELECT titolo FROM Film F1, Film F2
WHERE F1.valutaz = F2.valutaz AND
      F2.titolo = 'le iene' AND
      F1. titolo <> F2. titolo;
```



# Sotto-interrogazioni

- E' possibile per una sotto-interrogazione avere al suo interno un'altra sotto-interrogazione, predicati di join e tutti i predicati visti fino ad ora
- La clausola di qualificazione di una interrogazione può contenere una qualsiasi combinazione di condizioni normali e condizioni con sotto-interrogazioni
- Nelle sotto-interrogazioni scalari se nessuna tupla verifica la sotto-interrogazione, viene restituito il valore NULL

# Subquery scalare



- Tramite il meccanismo delle sotto-interrogazioni è possibile esprimere interrogazioni anche più complesse
- Esempio: determinare i film la cui valutazione è superiore alla media:



# Subquery scalare



- Tramite il meccanismo delle sotto-interrogazioni è possibile esprimere interrogazioni anche più complesse
- Esempio: determinare i film la cui valutazione è superiore alla media:

```
SELECT * FROM Film  
WHERE valutaz > (SELECT AVG(valutaz)  
                  FROM Film);
```

- La sotto-interrogazione restituisce il valore 3.55, quindi solo i film con valutazione superiore a tale valore vengono selezionati

# Esempio



Risultato:

titolo	regista	anno	genere	valutaz
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
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
le iene	quentin tarantino	1992	thriller	4.00



# Subquery scalare

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

```
SELECT titolo, regista FROM Film
WHERE genere = 'drammatico' AND
      valutaz = (SELECT MIN(valutaz)
                  FROM Film
                  WHERE genere = 'drammatico');
```

- È possibile definire una formulazione equivalente con il join?



## 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



# Massimo e subquery

La persona (o le persone) con il reddito massimo:

```
SELECT *  
FROM Persone  
WHERE Reddito = (SELECT MAX(Reddito)  
                  FROM Persone);
```



# Sotto-interrogazioni

- Negli esempi visti, le sotto-interrogazioni sono **scalari**
- Se una sotto-interrogazione scalare restituisce più di un valore si genera un errore a run-time
- Se si vuole invece utilizzare una sotto-interrogazione che restituisce più valori è necessario specificare come i valori restituiti devono essere usati nella clausola WHERE

# Sotto-interrogazioni di colonna



- Sintassi:
  - Operatore di confronto (ANY | ALL) sottoQuery
  - ANY: il predicato è vero se almeno uno dei valori restituiti da sottoQuery soddisfa la condizione. Restituisce FALSE se la sotto-interrogazione non restituisce tuple
  - ALL: il predicato è vero se tutti i valori restituiti dalla sottoQuery soddisfano la condizione. Restituisce TRUE se la sotto-interrogazione non restituisce tuple



# Esempio

- Determinare titolo, regista ed anno dei film più vecchi di **tutti i** film di Quentin Tarantino (si può fare anche con MIN):

SELECT titolo, regista, anno

FROM Film

WHERE anno < **ALL** (SELECT anno FROM Film

WHERE regista = 'quentin tarantino');

- Risultato (il primo film di Quentin Tarantino è del 1992):

titolo	regista	anno
edward mani di forbice	tim burton	1990
mediterraneo	gabriele salvatores	1991





# Sotto-interrogazioni

- In generale, nella formulazione di interrogazioni di massimo e di minimo:
  - $\geq$  ALL è equivalente a MAX
  - $\leq$  ALL è equivalente a MIN

# Sotto-interrogazioni



Determinare il titolo ed il regista del film drammatico di valutazione minima (con  $\leq$  ALL invece di MIN):

```
SELECT titolo, regista FROM Film
WHERE genere = 'drammatico' AND
      valutaz  $\leq$  ALL (SELECT valutaz
                      FROM Film
                      WHERE genere = 'drammatico');
```



## 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

- Nome e reddito dei padri di persone che guadagnano più di 20:
  - Formulazione con subquery alternativa al join che abbiamo già visto



## Con il join

Nome e reddito dei padri di persone che guadagnano più di 20:

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



# Esempio

- Nome e reddito dei padri di persone che guadagnano più di 20:
  - Scomposizione del problema in più sottoproblemi:
    - nome delle persone che guadagnano più di 20 (Q1)
    - nome e reddito dei padri di persone in Q1

# Q1



Nome delle persone che guadagnano più di 20:

## Persone

```
SELECT Nome  
FROM Persone  
WHERE Reddito > 20;
```

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



Nome e reddito dei padri di persone in Q1:



### 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



Nome e reddito dei padri:

```
SELECT Padre, Reddito  
FROM Paternita JOIN Persone ON (Padre = Nome)
```

# Esempio



Nome e reddito dei padri di persone in Q1:

```
SELECT Padre, Reddito
FROM Paternita JOIN Persone ON (Padre = Nome)
WHERE Figlio = ANY (SELECT Nome
                     FROM Persone
                     WHERE Reddito > 20);
```

# Any, ALL



- Sono definite le seguenti abbreviazioni per ANY ed ALL:
  - **IN** equivalente ad **= ANY**
  - **NOT IN** equivalente a **≠ ALL**

# Operatore NOT IN



- Concetto di esclusione:
  - Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton



## 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	?

## Cliente

codCli	nome	cognome	telefono	dataN	residenza
6610	anna	rossi	01055664433	05-Ott-1979	via scribanti 16 16131 genova
6635	paola	bianchi	0104647992	12-Apr-1976	via dodecaneso 35 16146 genova
6642	marco	verdi	3336745383	16-Ott-1972	via lagustena 35 16131 genova



# Concetto di esclusione

- Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton
- E possibile esprimere l'interrogazione mediante un join?

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista <> 'tim burton';
```



# Concetto di esclusione

- A che interrogazione corrisponde?

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista <> 'tim burton';
```





# Concetto di esclusione

- Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton:
  - occorre escludere dai clienti:
    - i clienti che hanno noleggiato un film di Tim Burton



# Concetto di esclusione

Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton:

```
SELECT codCli  
FROM Cliente  
WHERE codCli
```



?

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista = 'tim burton'
```



# Concetto di esclusione

Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton:

```
SELECT codCli  
FROM Cliente  
WHERE codCli NOT IN (SELECT codCli  
                      FROM Noleggio NATURAL JOIN Video  
                      WHERE regista = 'tim burton');
```



# Equivalenza con differenza

Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton:

```
SELECT codCli  
FROM Cliente
```

**EXCEPT**

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

# Concetto di esclusione



Trovare il codice dei clienti che hanno noleggiato  
**solo** film di Tim Burton



## 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	?



# Concetto di esclusione

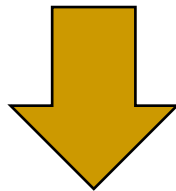
- A che interrogazione corrisponde?

~~SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista = 'tim burton';~~



# Concetto di esclusione

Trovare il codice dei clienti che hanno noleggiato **solo** film di Tim Burton:



Trovare i clienti che hanno noleggiato film di Tim Burton e  
non hanno mai noleggiato film  
di registi diversi

- Insieme da escludere:
  - clienti che hanno noleggiato film non di Tim Burton



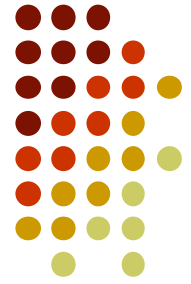
# Concetto di esclusione



Trovare il codice dei clienti che hanno noleggiato **solo** film di Tim Burton:

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista <> 'tim burton'
```

Clienti che hanno  
noleggiato  
almeno un film  
non di TB



# Concetto di esclusione

Trovare il codice dei clienti che hanno noleggiato **solo** film di Tim Burton:

```
SELECT codCli
FROM Noleggio NATURAL JOIN Video
WHERE regista = 'tim burton' AND
      codCli NOT IN ( SELECT codCli
                      FROM Noleggio NATURAL JOIN Video
                      WHERE regista <> 'tim burton');
```



# Equivalenza con differenza

Trovare il codice dei clienti che hanno noleggiato **solo** film di Tim Burton:

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista = 'tim burton'
```

**EXCEPT**

```
SELECT codCli  
FROM Noleggio NATURAL JOIN Video  
WHERE regista <> 'tim burton';
```

# Subquery & clausola HAVING



**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)

Il dipartimento che spende il massimo in stipendi:

- Quanto spende in totale ogni dipartimento per lo stipendio dei suoi impiegati
- Trovare quello che corrisponde al massimo

# Subquery & clausola HAVING



**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)

Il dipartimento che spende il massimo in stipendi:

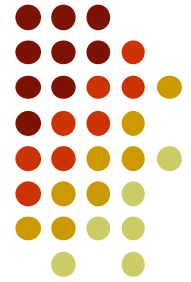
- Quanto spende ogni dipartimento per lo stipendio dei suoi impiegati

```
SELECT SUM(Stipendio)
```

```
FROM Impiegati
```

```
GROUP BY Dip;
```

# Subquery & clausola HAVING



**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)

Il dipartimento che spende il massimo in stipendi:

```
SELECT Dip
FROM Impiegati
GROUP BY Dip
HAVING SUM(Stipendio) >= ALL(SELECT SUM(Stipendio)
                               FROM Impiegati
                               GROUP BY Dip);
```

Impiegato(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
Dipartimento(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
Collocazione\_Dip(IdD, Luogo)  
Progetto(NomeP, IdP, Luogo, Dip)  
Assegnato(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



- Q1: id e numero degli impiegati che guadagnano più di 40.000 euro, per ogni dipartimento con più di 5 impiegati
- Q2 Il nome degli impiegati che non lavorano in un progetto controllato dal dip 5

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q1: id e numero degli impiegati che guadagnano più di 40.000 euro, per ogni dipartimento con più di 5 impiegati

- Sottoproblemi:
  - dipartimenti con più di 5 impiegati
  - per quei dipartimenti: id e numero di impiegati che guadagnano più di 40.000 euro



Impiegato(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
Dipartimento(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
Collocazione\_Dip(IdD, Luogo)  
Progetto(NomeP, IdP, Luogo, Dip)  
Assegnato(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Dipartimenti con più di 5 impiegati:

**Q11:**  
**SELECT Dip**  
**FROM Impiegato**  
**GROUP BY Dip**  
**HAVING COUNT(\*) > 5**

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



per ogni dipartimento nel risultato di Q11, id e numero degli impiegati  
che guadagnano più di 40000 euro

```
SELECT Dip, COUNT(*) AS Numero_Imp  
FROM Impiegato  
GROUP BY Dip;
```

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q1: per ogni dipartimento nel risultato di Q11, id e numero degli impiegati che guadagnano più di 40000 euro

```
SELECT Dip, COUNT(*) AS Numero_Imp  
FROM Impiegato  
WHERE .....  
GROUP BY Dip;
```

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q1: per ogni dipartimento nel risultato di Q11, id e numero degli impiegati che guadagnano più di 40000 euro

```
SELECT Dip, COUNT(*) AS Numero_Imp  
FROM Impiegato  
WHERE Stipendio > 40000 AND Dip IN Q11  
GROUP BY Dip;
```

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q1: per ogni dipartimento nel risultato di Q11, id e numero degli impiegati che guadagnano più di 40000 euro

```
SELECT Dip, COUNT(*) AS Numero_Imp  
FROM Impiegato  
WHERE Stipendio > 40000  
      AND Dip IN (SELECT Dip  
                  FROM Impiegato  
                  GROUP BY Dip  
                  HAVING COUNT(*) > 5)  
  
GROUP BY Dip;
```



**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)

Q1: id e numero degli impiegati che guadagnano più di 40.000 euro, per ogni dipartimento con più di 5 impiegati

```
SELECT Dip, COUNT(*) AS Numero_Imp  
FROM Impiegato  
WHERE Stipendio > 40000  
GROUP BY Dip  
HAVING COUNT(*) > 5;
```

Perchè questa query è sbagliata?

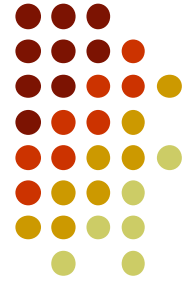
**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q2 Il nome degli impiegati che non lavorano in un progetto controllato dal dip 5

- Sottoproblemi:
  - progetti controllati dal dip 5
  - nome degli impiegati che non lavorano in quei progetti

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Progetti controllati dal dip 5

```
SELECT IdP  
FROM Progetto  
WHERE Dip = 5;
```



**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



CF degli impiegati che lavorano a progetti controllati dal dip 5

```
SELECT CF  
FROM Assegnato  
WHERE IdP IN (SELECT IdP FROM Progetto  
                WHERE Dip = 5);
```

**Impiegato**(Nome, CF, Indirizzo, Stipendio, Resp<sup>Impiegato</sup>, Dip<sup>Dipartimento</sup>)  
**Dipartimento**(Nome, IdD, Manager<sup>Impiegato</sup>, ManagerData\_In)  
**Collocazione\_Dip**(IdD, Luogo)  
**Progetto**(NomeP, IdP, Luogo, Dip)  
**Assegnato**(CF<sup>Impiegato</sup>, IdP<sup>Progetto</sup>, ore)



Q2 Il nome degli impiegati che non lavorano in un progetto controllato dal dip 5

```
SELECT Nome
FROM Impiegato
WHERE CF NOT IN (SELECT CF
                  FROM Assegnato
                  WHERE IdP IN (SELECT IdP FROM Progetto
                               WHERE Dip = 5));
```



# Subquery di tabella

- Una sotto-interrogazione può restituire come risultato una tabella con più di due attributi:
  - nel predicato di confronto è necessario usare il **costruttore di tupla**
  - permette di definire la struttura temporanea di una tupla
    - si elencano gli attributi che ne fanno parte tra ()
      - (NomeAttributo, NomeAttributo, ..)
    - estende il potere espressivo di IN, NOT IN



# Esempio

- Viaggio(CodV,Partenza,Arrivo,OraP,OraA)
- Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di due ore:

```
SELECT Partenza, Arrivo  
FROM Viaggio  
WHERE OraA – OraP >2
```

Coppie partenza/arrivo per  
cui esistono viaggi che  
durano più di due ore



# Esempio

- Viaggio(CodV,Partenza,Arrivo,OraP,OraA)
- Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di due ore:

```
SELECT Partenza, Arrivo  
FROM Viaggio  
WHERE (Partenza, Arrivo) NOT IN
```

Costruttore di tupla

```
(SELECT Partenza, Arrivo  
FROM Viaggio  
WHERE OraA – OraP >2);
```



# Sotto-interrogazioni correlate

- Negli esempi visti fino ad ora ogni sotto-interrogazione veniva **eseguita una volta per tutte** ed il valore (o l'insieme di valori) usato nella clausola WHERE della interrogazione esterna
- Nelle sotto-interrogazioni correlate la sotto-interrogazione viene **eseguita ripetutamente** per ogni **tupla candidata** considerata nella valutazione della interrogazione esterna
- La condizione di correlazione lega l'esecuzione dell'interrogazione interna al valore di attributi della tupla corrente nell'interrogazione esterna



# Esempio

- Determinare titolo, regista ed anno dei film la cui valutazione è superiore alla media delle valutazioni dei film **dello stesso regista:**

# 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



# Sotto-interrogazioni correlate



- E' necessaria una interrogazione esterna che selezioni i film in base ad un predicato sulla valutazione:

SELECT titolo, regista, anno FROM Film

WHERE valutaz > (*media delle valutazioni dei film  
del regista del film candidato*);

- E' inoltre necessaria una sotto-interrogazione che calcoli la media delle valutazioni dei film del regista di ogni tupla candidata:

(SELECT AVG(valutaz) FROM Film

WHERE regista=(*valore di regista nella tupla  
candidata*));

# Sotto-interrogazioni correlate



- Determinare titolo, regista ed anno dei film la cui valutazione è superiore alla media delle valutazioni dei film **dello stesso regista**:

```
SELECT titolo, regista, anno FROM Film X
WHERE valutaz > (SELECT AVG(valutaz)
                 FROM Film
                 WHERE regista = X.regista);
```

- X è un alias per la relazione Film

# Sotto-interrogazioni correlate



```
SELECT titolo, regista, anno FROM Film X
WHERE valutaz > (SELECT AVG(valutaz)
                 FROM Film
                 WHERE regista = X.regista);
```

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



- Risultato:

titolo	regista	anno
edward mani di forbice	tim burton	1990
nightmare before christmas	tim burton	1993
ed wood	tim burton	1994
la fabbrica di cioccolato	tim burton	2005
io non ho paura	gabriele salvatores	2003
mediterraneo	gabriele salvatores	1991
le iene	quentin tarantino	1992

# Sotto-interrogazioni correlate

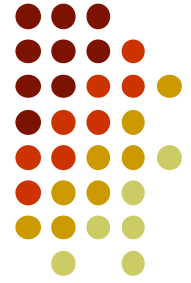


- Ogni volta che l'interrogazione esterna considera una tupla candidata, deve invocare la sotto-interrogazione e "passare" il regista del film in esame
- La sotto-interrogazione calcola quindi la media delle valutazioni dei film del regista che è stato passato e restituisce tale valore all'interrogazione esterna
- L'interrogazione esterna può quindi confrontare la valutazione del film in esame con il valore restituito dalla sotto-interrogazione

# Sotto-interrogazioni correlate



- Questo tipo di interrogazioni è chiamato **correlato**, perchè ogni esecuzione della sotto-interrogazione è correlata al valore di uno o più attributi delle tuple candidate nella interrogazione principale
- Per poter fare riferimento alle colonne delle tuple candidate nella interrogazione esterna si fa uso degli **alias di relazione**
- L'alias di relazione è definito nell'interrogazione esterna e riferito nella sotto-interrogazione correlata
- Non è possibile far riferimento a alias definiti in blocchi più interni



# Esempio

- Viaggio(CodV,Partenza,Arrivo,OraP,OraA)
- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)



# Esempio

Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo):

```
SELECT CodV  
FROM Viaggio AS VA  
WHERE OraA – OraP < (....  
.....);
```

durata media dei viaggi  
sul percorso corrente

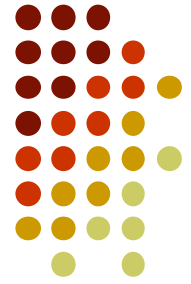




# Esempio

Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo):

```
SELECT CodV  
FROM Viaggio AS VA  
WHERE OraA – OraP <  
  (SELECT AVG(OraA – OraP) FROM Viaggio  
   );
```



# Esempio

Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo):

```
SELECT CodV
FROM Viaggio AS VA
WHERE OraA – OraP <
  (SELECT AVG(OraA – OraP) FROM Viaggio
   WHERE Partenza = VA.Partenza AND
    Arrivo = VA.Arrivo);
```

**Condizione di correlazione**



# EXIST e NOT EXISTS

- Una sotto-interrogazione sq può essere usata in combinazione con gli operatori EXISTS e NOT EXISTS
  - **EXISTS(sq)** restituisce il valore Booleano TRUE se sq restituisce almeno una tupla; restituisce FALSE altrimenti
  - **NOT EXISTS(sq)** restituisce il valore Booleano TRUE se sq non restituisce alcuna tupla; restituisce FALSE altrimenti

### 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

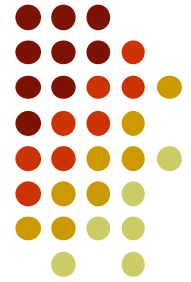


# Esempio



Il nome e l'età delle persone che hanno almeno un figlio:

# Esempio



Il nome e l'età delle persone che hanno almeno un figlio:

```
SELECT Nome, Età FROM Persone P
WHERE EXISTS (SELECT * FROM PATERNITA
              WHERE Padre = P.Nome) OR
EXISTS (SELECT * FROM MATERNITA
        WHERE Madre = P.Nome);
```

## 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

## Paternità

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

- Il predicato EXISTS è falso per Andrea poiché non esistono suoi figli nella tabella Maternità/Paternità
  - Andrea non fa parte del risultato dell'interrogazione





## 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

## Paternità

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

- Il predicato EXISTS è vero per Maria poiché esistono suoi figli nella tabella Maternità
  - Maria fa parte del risultato dell'interrogazione



# Esempio

Risultato:

Nome	Età
Maria	55
Anna	50
Luigi	50
Franco	60
Sergio	85
Luisa	75



# EXISTS e NOT EXISTS



- Nell'interrogazione interna a EXISTS/NOT EXISTS, la clausola SELECT è obbligatoria, ma irrilevante, perchè gli attributi non sono visualizzati

# Concetto di esclusione



Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton



## Noleggio

### 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

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	?

### Cliente

codCli	nome	cognome	telefono	dataN	residenza
6610	anna	rossi	01055664433	05-Ott-1979	via scribanti 16 16131 genova
6635	paola	bianchi	0104647992	12-Apr-1976	via dodecaneso 35 16146 genova
6642	marco	verdi	3336745383	16-Ott-1972	via lagustena 35 16131 genova



# Concetto di esclusione

Trovare il codice dei clienti che **non** hanno noleggiato film di Tim Burton:

```
SELECT codCli
FROM Cliente C
WHERE NOT EXISTS( SELECT *
                  FROM Noleggio NATURAL JOIN Video
                  WHERE regista = 'tim burton'
                  AND codCli = C.codCli);
```



# Esempio

- Studenti(Matricola,Nome,Cognome)
- Trovare tutti gli studenti che NON hanno un omonimo:
  - ```
SELECT *  
FROM Studenti S  
WHERE NOT EXISTS (SELECT *  
FROM Studenti S2  
WHERE S2.Nome = S.Nome  
AND S2.Cognome = S.Cognome  
AND S2.Matricola <> S.Matricola);
```



# Intersezione e differenza

- Le operazioni di intersezione e differenza possono essere eseguite tramite EXISTS e NOT EXISTS
- Esempio:
  - Determinare gli anni in cui sono usciti sia film di Tim Burton sia film di Quentin Tarantino:

```
SELECT DISTINCT anno
FROM Film WHERE regista = 'tim burton'
INTERSECT
SELECT DISTINCT anno FROM Film
WHERE regista = 'quentin tarantino';
```

# 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    |





# Intersezione e differenza

- Oppure:

```
SELECT DISTINCT anno FROM Film F
WHERE regista = 'tim burton' AND
      EXISTS (SELECT * FROM Film
              WHERE regista = 'quentin tarantino'
              AND anno = F.anno);
```



# Intersezione e differenza

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

```
SELECT DISTINCT anno
FROM Film WHERE regista = 'tim burton'
EXCEPT
SELECT DISTINCT anno FROM Film
WHERE regista = 'quentin tarantino';
```

# Esempio



- Oppure:

```
SELECT DISTINCT anno FROM Film F
WHERE regista = 'tim burton' AND
      NOT EXISTS (SELECT * FROM Film
                  WHERE regista = 'quentin tarantino'
                  AND anno = F.anno);
```

# Quantificatore universale



- le sotto-interrogazioni correlate insieme all'operatore NOT EXISTS permettendo anche di esprimere il quantificatore universale

# 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    |
|                             | quentin tarantino   | 1992 | thriller     | 4.00    |

## Video

| colloc | titolo                      | regista             | tipo | Noleggio | colloc | dataHol     | codCli | dataRest    |
|--------|-----------------------------|---------------------|------|----------|--------|-------------|--------|-------------|
| 1111   | underground                 | emir kusturica      | v    |          | 1111   | 01-Mar-2006 | 6635   | 02-Mar-2006 |
| 1112   | underground                 | emir kusturica      | d    |          | 1115   | 01-Mar-2006 | 6635   | 02-Mar-2006 |
| 1113   | big fish                    | tim burton          | v    |          | 1117   | 02-Mar-2006 | 6635   | 06-Mar-2006 |
| 1114   | big fish                    | tim burton          | d    |          | 1118   | 02-Mar-2006 | 6635   | 06-Mar-2006 |
| 1115   | edward mani di forbice      | tim burton          | d    |          | 1111   | 04-Mar-2006 | 6642   | 05-Mar-2006 |
| 1116   | nightmare before christmas  | tim burton          | v    |          | 1119   | 08-Mar-2006 | 6635   | 10-Mar-2006 |
| 1117   | nightmare before christmas  | tim burton          | d    |          | 1120   | 08-Mar-2006 | 6635   | 10-Mar-2006 |
| 1118   | ed wood                     | tim burton          | d    |          | 1116   | 08-Mar-2006 | 6642   | 09-Mar-2006 |
| 1119   | mars attacks                | tim burton          | d    |          | 1118   | 10-Mar-2006 | 6642   | 11-Mar-2006 |
| 1120   | il mistero di sleepy hollow | tim burton          | d    |          | 1121   | 15-Mar-2006 | 6635   | 18-Mar-2006 |
| 1121   | la sposa cadavere           | tim burton          | d    |          | 1122   | 15-Mar-2006 | 6635   | 18-Mar-2006 |
| 1122   | la fabbrica di cioccolato   | tim burton          | d    |          | 1113   | 15-Mar-2006 | 6635   | 18-Mar-2006 |
| 1123   | la fabbrica di cioccolato   | tim burton          | d    |          | 1129   | 15-Mar-2006 | 6635   | 20-Mar-2006 |
| 1124   | io non ho paura             | gabriele salvatores | d    |          | 1119   | 15-Mar-2006 | 6642   | 16-Mar-2006 |
| 1125   | nirvana                     | gabriele salvatores | d    |          | 1126   | 15-Mar-2006 | 6610   | 16-Mar-2006 |
| 1126   | mediterraneo                | gabriele salvatores | d    |          | 1112   | 16-Mar-2006 | 6610   | 18-Mar-2006 |
| 1127   | pulp fiction                | quentin tarantino   | v    |          | 1114   | 16-Mar-2006 | 6610   | 17-Mar-2006 |
| 1128   | pulp fiction                | quentin tarantino   | d    |          | 1128   | 18-Mar-2006 | 6642   | 20-Mar-2006 |
| 1129   | le iene                     | quentin tarantino   | d    |          | 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

- Determinare i codici dei clienti che hanno noleggiato **tutti i film** di Tim Burton
- Risolvere questa interrogazione richiede di ragionare **in base al concetto di controesempio**
- Viene “riformulata” come:
  - **determinare i codici dei clienti per cui non esiste un film di Tim Burton che il cliente non ha mai noleggiato**

# Esempio



```
SELECT DISTINCT codCli FROM Noleggio X  
WHERE NOT EXISTS (film di Tim Burton che il cliente in esame  
non ha mai noleggiato);
```

# Esempio



```
SELECT DISTINCT codCli FROM Noleggio X
WHERE NOT EXISTS (SELECT * FROM Film F
                  WHERE regista = 'tim burton' AND
                  (non esiste un noleggio del film
                   considerato per il cliente in esame) );
```



# Esempio



```
SELECT DISTINCT codCli FROM Noleggio X
WHERE NOT EXISTS (SELECT * FROM Film F
                  WHERE regista = 'tim burton' AND
                  NOT EXISTS (SELECT *
                             FROM Noleggio NATURAL JOIN Video
                             WHERE codCli = X.codCli
                                AND titolo = F.titolo
                                AND regista = F.regista));
```

# Esempio



- Un modo alternativo di esprimere la query precedente è mediante l'uso di funzioni di gruppo

# Esempio



- Determinare i codici dei clienti che hanno noleggiato **tutti i film** di Tim Burton
- Osservazione:
  - per determinare i clienti che hanno noleggiato tutti i film di Tim Burton confrontiamo:
    - il numero di film di Tim Burton presenti nella videoteca
    - il numero di film di Tim Burton noleggiati da ogni cliente

# Esempio

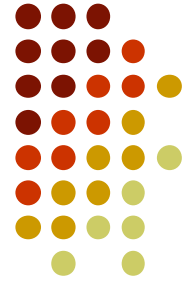


I codici dei clienti che hanno noleggiato **tutti i film** di Tim Burton:

```
SELECT COUNT(*)  
FROM Film  
WHERE regista = 'tim burton';
```

il numero di film  
di TB in catalogo

# Esempio

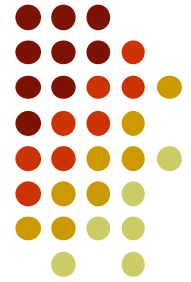


I codici dei clienti che hanno noleggiato **tutti i film** di Tim Burton:

```
SELECT codCli FROM Noleggio NATURAL JOIN Video  
WHERE regista = 'tim burton'  
GROUP BY codCli
```

Per ogni cliente,  
i film di TB che ha  
noleggiato

# Esempio



I codici dei clienti che hanno noleggiato **tutti i film** di Tim Burton:

```
SELECT codCli FROM Noleggio NATURAL JOIN Video
WHERE regista = 'tim burton'
GROUP BY codCli
HAVING COUNT(DISTINCT titolo) =
    (SELECT COUNT(*)
     FROM Film
     WHERE regista = 'tim burton');
```

# Esempio



- I padri i cui figli guadagnano tutti più di venti:

### 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      |



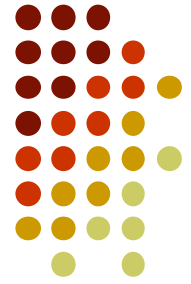




# Esempio

I padri i cui figli guadagnano tutti più di venti:

```
SELECT DISTINCT Padre
FROM Paternita Z
WHERE NOT EXISTS (
    SELECT *
    FROM Paternita W, Persone
    WHERE W.Padre = Z.Padre
        AND W.Figlio = Nome
        AND Reddito <= 20);
```



# Provate voi!

- Trovare gli studenti che hanno un omonimo:
  - ```
SELECT *  
FROM Studenti S  
WHERE (Nome, Cognome) IN  
      (SELECT Nome, Cognome  
       FROM Studenti S2  
       WHERE S2.Matricola <> S.Matricola);
```
  - Provare a risolverla: 1) non usando le sottoquery, 2) usando sottoquery + predicato EXISTS

DISCO(NroSerie, TitoloAlbum, Anno, Prezzo)  
CONTIENE(NroSerieDisco, CodiceReg, NroProg)  
ESECUZIONE(CodiceReg, TitoloCanz, Anno)  
AUTORE(Nome, TitoloCanzone)  
CANTANTE(NomeCantante, CodiceReg)



Determinare le chiavi esterne presenti nello schema

DISCO(NroSerie, TitoloAlbum, Anno, Prezzo)  
CONTIENE(NroSerieDisco, CodiceReg, NroProg)  
ESECUZIONE(CodiceReg, TitoloCanz, Anno)  
AUTORE(Nome, TitoloCanzone)  
CANTANTE(NomeCantante, CodiceReg)



Per come è definito lo schema:

1. Possono esistere registrazioni con lo stesso titolo di cantanti diversi?
2. Un cantante può cantare due volte la stessa canzone?
3. Un disco può contenere brani di autori diversi?
4. Un disco può contenere brani registrati in anni diversi?
5. Una canzone può essere cantata da due cantanti?



DISCO(NroSerie, TitoloAlbum, Anno, Prezzo)  
CONTIENE(NroSerieDisco, CodiceReg, NroProg)  
ESECUZIONE(CodiceReg, TitoloCanz, Anno)  
AUTORE(Nome, TitoloCanzone)  
CANTANTE(NomeCantante, CodiceReg)

Q1: I cantanti che hanno inciso brani nel 2009 e 2010:

```
SELECT NomeCantante
FROM CANTANTE JOIN ESECUZIONE USING(CodiceReg)
WHERE Anno = 2009
INTERSECT
SELECT NomeCantante
FROM CANTANTE JOIN ESECUZIONE ON USING(CodiceReg)
WHERE Anno = 2010;
```

DISCO(NroSerie, TitoloAlbum, Anno, Prezzo)  
CONTIENE(NroSerieDisco, CodiceReg, NroProg)  
ESECUZIONE(CodiceReg, TitoloCanz, Anno)  
AUTORE(Nome, TitoloCanzone)  
CANTANTE(NomeCantante, CodiceReg)



Q2: I cantanti che non hanno mai registrato una canzone come solisti:

```
SELECT DISTINCT NomeCantante
FROM CANTANTE
WHERE NomeCantante NOT IN
( SELECT NomeCantante
  FROM CANTANTE S1
  WHERE CodiceReg NOT IN
  (SELECT CodiceReg
   FROM CANTANTE S2
   WHERE S2.NomeCantante <> S1.NomeCantante ) );
```

DISCO(NroSerie, TitoloAlbum, Anno, Prezzo)  
CONTIENE(NroSerieDisco, CodiceReg, NroProg)  
ESECUZIONE(CodiceReg, TitoloCanz, Anno)  
AUTORE(Nome, TitoloCanzone)  
CANTANTE(NomeCantante, CodiceReg)



Q3: I cantautori puri (cioè i cantanti che hanno eseguito solo canzoni di cui erano autori)

```
SELECT NomeCantante
FROM CANTANTE
WHERE NomeCantante NOT IN (
    SELECT NomeCantante
    FROM CANTANTE JOIN ESECUZIONE USING(CodiceReg) JOIN AUTORI
        ON(TitoloCanz=TitoloCanzone)
    WHERE Nome <> NomeCantante);
```